IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 7, 2020, accepted June 16, 2020, date of publication June 23, 2020, date of current version July 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004536

A Novel and Efficient 3D Multiple Images Encryption Scheme Based on Chaotic Systems and Swapping Operations

MUHAMMAD HANIF[1,2,*], RIZWAN ALI NAQVI[3,*], (Member, IEEE), SAGHEER ABBAS[1], MUHAMMAD ADNAN KHAN[4], AND NADEEM IQBAL[1,5]

[1]Department of Computer Science, NCBA&E, Lahore 54660, Pakistan
[2]Department of Computer Science, Bahria University Lahore Campus, Lahore 54000, Pakistan
[3]Department of Unmanned Vehicle Engineering, Sejong University, Seoul 05006, South Korea
[4]Department of Computer Science, Lahore Garrison University, Lahore 54000, Pakistan
[5]School of Computing and Information Sciences, Imperial College of Business Studies, Lahore 53720, Pakistan

Corresponding author: Muhammad Adnan Khan (madnankhan@lgu.edu.pk)

*Muhammad Hanif and Rizwan Ali Naqvi are co-first authors.

This work was supported by the Sejong University Faculty Research Fund.

ABSTRACT Single image encryption schemes are not efficient enough when a bunch of images is to be encrypted in some real-world setting. To overcome this problem, an efficient and secured multiple images encryption scheme is proposed in this study using two chaotic maps and simple row and column swapping operations in a 3D image space. The N input images are piled to make a 3D image. To confuse the given pixel data, two images are chosen randomly from this pile. The randomly chosen two rows from the two randomly chosen images are swapped with each other. In the same way, two randomly chosen columns are swapped with each other. The operation of randomly chosen two images, two rows, and two columns have been iterated an arbitrary number of times to throw the confusion effects in the pixels data. Intertwining Logistic Map (ILM) and Improved Piecewise Linear Chaotic Map (MPWLCM) have been used to get the four streams of random numbers. The three streams of the former map have been used to create the confusion effects, whereas the fourth stream of random numbers given by the latter map has been used for the diffusion effects. SHA-256 hash codes have been used to throw the plaintext sensitivity in the proposed cipher. Besides, a 256-bit user key has been employed to increase the key space. Both the simulation and the exhaustive security analyses carried out at the end vividly prove the security, resistance to the varied attacks, and the real-world applicability of the proposed cipher.

INDEX TERMS Image processing, chaos, encryption, decryption, cryptography, 3D.

I. INTRODUCTION

In this modern age, cameras are embedded in virtually every digital device. The resolution of these digital devices is getting high with each passing day. These digital "eyes" are now capable to digitally capture from tiny microscopic particles to gigantic galaxies in the form of images. These digital images are now part of our daily lives, where these images are used for a lot of purposes. From family pictures to the digital blueprint of an advanced lab, these images play a vital role in our daily routine life. As the digital world is growing, the security and privacy aspects are also increasing. The protection of these digital images from some potential antagonists and adversary either at some storage or during transmission is one of the very demanding features of this modern world. The traditional data encryption algorithms DES, AES, RSA, etc. cannot work over the digital images. The reason behind this is that these algorithms are specifically designed for textual data and most of them are working on blocks with fixed sizes, *i.e.*, 64-bit, 128-bit. In sharp contrast to that, digital images are composed of picture elements, *i.e.*, pixels. The digital images have their properties and structure like strong correlation among the adjacent pixels, high volume, and redundancy. These features hinder the application of the above mentioned traditional textual encryption algorithms over the digital images [1], [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

In image encryption, random streams have great importance [3]–[10]. These streams are generated using different chaotic maps. Depending upon the chaotic map capabilities, several streams are produced. Some maps can produce a single chaotic stream, e.g., peace wise linear chaotic map [11] and some can produce more; intertwining logistic maps can produce three chaotic streams [12], for example. Those chaotic maps that produce one or two streams are normally called low-dimensional chaotic maps [13]–[15], and the ones which produce more than two chaotic streams are called high-dimensional chaotic maps [16]–[18]. Although low-dimensional chaotic maps are easy to implement, they sometimes do not fulfill the requirement of the requisite randomness required for the modern smart ciphers. On the other hand, high-dimensional maps are difficult to implement and are time-consuming as well but they provide relatively more random data streams. These streams are pseudo-random and depend on the initial values and the system's parameters of the concerned chaotic map. These properties are very promising for the enterprise of cryptography due to which researchers use these maps in their schemes for random data generation. These streams play a vital role in the encryption process and provide help in the permutation (confusion/scrambling) and substitution (diffusion) process. These streams also help to safeguard against attacks like plaintext-attack, differential-attack, chosen-plaintext attack, and cipher-attack, etc. In our proposed scheme, we have used two chaotic maps, *i.e.*, Intertwining-Logistic Map (ILM) and Improved-Piecewise Linear-Chaotic Map (MPWLCM) to fulfill the requirements of chaoticity. A total of four random/chaotic streams were generated; three from ILM and one from MPWLCM that were used to fulfill the algorithmic logic and to perform the encryption/decryption tasks.

In the recent past, dozens of image encryption schemes have been developed, some strong and some weak. In these schemes, the majority are Single Image Encryption schemes (SIE) and very few are Multiple Images Encryption schemes (MIE). In the literature, one can easily find encryption schemes that only perform the permutation (confusion/scrambling) [19]–[22]. These schemes are prone to attacks and can easily be cracked using different types of attacks, e.g., brute force attack, differential-attack, chosen-plaintext attack, etc. On the other hand, most ciphers are composed of both permutation (confusion) and substitution (diffusion) [7], [8], [12], [23]–[25]. This not only strengthens the cipher but also provides extra security to safeguard against the attacks.

The MIE schemes are gaining attention day by day. Many MIE schemes are presented in the literature [26]–[31]. In [29], an MIE was developed using the DNA and index-based permutation and diffusion by joining the input images into a large single image. This image was then transformed into a one-dimensional array, which was sub-divided into two halves. In the permutation process, indexes of the array were used for scrambling the pixels. The same indexes were then used for the substitution process which was based on the DNA. In another MIE scheme [26], the authors grouped grayscale images into non-overlapping blocks and performed permutation and substitution operations over the blocks using PWLCM. A yet another MIE scheme was introduced by [27] using a two-dimensional chaotic economic map. They grouped the images and split them into its pure image elements. The permutation was achieved via merging the images into one large image and then using the previously mentioned map, they got the elements of the mixed image. This large cipher image was then broken into smaller images. In [28], the authors developed an MIE scheme based on DNA and chaotic system. The SHA-256 hash value was used to update the initial parameter of the chaotic system. They merged the input images into a large image and scrambled the pixels using a chaotic map. The substitution was done using DNA and XOR operation. In another MIE scheme [30], they used the phase mask multiplexing. But the main issue with this scheme was that as the input images got increased, the quality of the decrypted images degraded. Some MIE schemes [30], [32] are limited to several images. These encryption schemes encrypt four images in one session, which is a limitation over the MIE. In [31], they segmented the input images into its elements, and then scrambled all the elements using a chaotic map. Lastly, a simple XOR operation was carried out to realize the diffusion effects.

In the literature, many encryption schemes based on different swapping techniques exist [33]–[39] targeting single images only. These techniques vary from one another in their design principles. The majority of these techniques perform the swapping operation on single-pixel only. For instance, in [33], the permutation process was carried out using pixel swapping sequences comprising eight-pixels permutation sequences, eight masking sequences, and lastly, sixteen pixels swapping sequences were generated. In the end, the substitution process was performed using the XOR operation with the masking sequence. A very simple swapping mechanism was adopted in [34]. They shuffled the input image pixels using the 2D standard map; then they substituted the pixels' values using the 2D lookup tables. This process was conducted for a significant amount of time to achieve satisfactory results. In yet another scheme presented in [35], the chaotic logistic map was used for the pixels swapping. In this scheme, the adjacent pixels were swapped in the permutation process. But the main limitation plaguing this process was the lack of randomization since the pixels got swapped with the adjacent pixels only. This scheme can be improved if each pixel has the same probability to swap with any other pixel in the input image. In [36], swapping operation over pixels was implemented using the three chaotic tent maps. In still another scheme [37], block-level swapping was used. To achieve time efficiency, they used block-level swapping instead of pixel-level swapping. Although they achieved the time efficiency up to some extent the security of the cipher was compromised. In another study conducted by [38], they used the pixel swapping for the confusion. Again, the pixels got swapped next to each other which narrowed the randomization process resulting in the curtailment of the security effects. The reason was that each pixel in the image did not have an equal chance for swapping with any other pixel in the image. In an image cipher [39], Henon map was used for the generation of chaotic data. This map selected a random pixel from the image to swap sequentially with the other pixels.

In the proposed project, we present a novel scheme for confusion/scrambling. After the images are stacked to form a 3D image, two images are selected arbitrarily from this pile. From these two images, two rows are selected randomly from each image and are swapped with each other. The same happens for the columns. There is no constraint for the rows and columns and they are swapped freely in all the input images. In this way, the pixel data is inter-blended abundantly which results in the improvement of the validation metrics. The following bullet points characterize the contribution of this study.

- The input grayscale images are stacked to form a 3D image before the confusion and diffusion operations are launched. The cryptanalysis of such ciphers becomes more difficult as compared to their 2D counterparts.
- Through a very simple method of rows swapping and columns swapping across random images in the 3D image, the permutation effects have been achieved. Although this process sounds very simple on the surface, it has very far-reaching implications as far as scrambling is concerned. It shifts the entire row and entire column from one random image to another random image via swapping.
- The 256-bits hash key and 256-bits user key have been used to achieve the plaintext sensitivity, *i.e.*, for every input image even with a single bit change will produce a drastically different cipher image. Besides, these keys also cause to increase in the key space - a deterrent to counter any brute force threat from the cryptanalysis community.
- The proposed scheme can be easily tailored for multiple RGB images encryption and decryption.

The plan for the remaining paper has been set as follows: Section 2 discusses the chaotic maps and the swapping operation for the rows and columns. Section 3 talks about the key generation, initial parameters generation, chaotic streams generation, encryption/decryption procedures of the proposed scheme. Section 4 presents the simulation of the proposed scheme. Section 5 gives a detailed security analysis. The paper ends by giving the concluding remarks in the last Section 6.

II. PRELIMINARIES

A. CHAOTIC MAPS

To obtain random effects for conducting confusion and diffusion processes on the images, chaotic maps have been used by the researchers. A chaotic map is a random number stream generator with unpredictable values. In this paper, two chaotic maps have been used to obtain random numbers. These maps are Intertwining Logistic Map (ILM) and Improved Piecewise Linear Chaotic Map (MPWLCM). The ILM renders three streams that have been used for realizing the confusion effects, whereas MPWLCM has a single stream that has been used for creating the diffusion effects in the proposed cipher.

1) INTERTWINING LOGISTIC MAP (ILM)

A typical example of a three-dimension chaotic-map is Intertwining Logistic Map (ILM) which is the refined form of the 1D and 2D logistic maps. This map has a larger key space as defined below (1):

$$\begin{cases} x_{n+1} = [\mu \times k_1 \times y_n \times (1 - x_n) + z_n] \, mod\, 1 \\ y_{n+1} = \left[\mu \times k_2 \times y_n + z_n \times {}^{1}\!/\!(1 + x_{n+1}^2)\right] mod\, 1 \\ z_{n+1} = [\mu \times (x_{n+1} + y_{n+1} + k_3) \times sin z_n] \, mod\, 1 \end{cases} \tag{1}$$

where $0 < \mu \leq 3.999$, $|k_1| > 33.50$, $|k_2| > 37.97$, $|k_3| > 35.7$. As compared to the logistic map, this map generates better chaotic behavior with no blank windows and has significant even distribution [12].

2) IMPROVED PIECEWISE LINEAR CHAOTIC MAP (MPWLCM)

Improved Piecewise Linear Chaotic Map (MPWLCM) is a single stream chaotic map [11]. The MPWLCM is represented by (2):

$$b_{a+1} = F(b_a, c) = \frac{b_a - \left\lfloor {}^{b_a}\!/\!_c \right\rfloor \times c}{c} \tag{2}$$

where $b_a \in (0, 1)$, and c is the control parameter and its value will be from 0 to 0.5. The resulting values of b are between 0 and 1 [11].

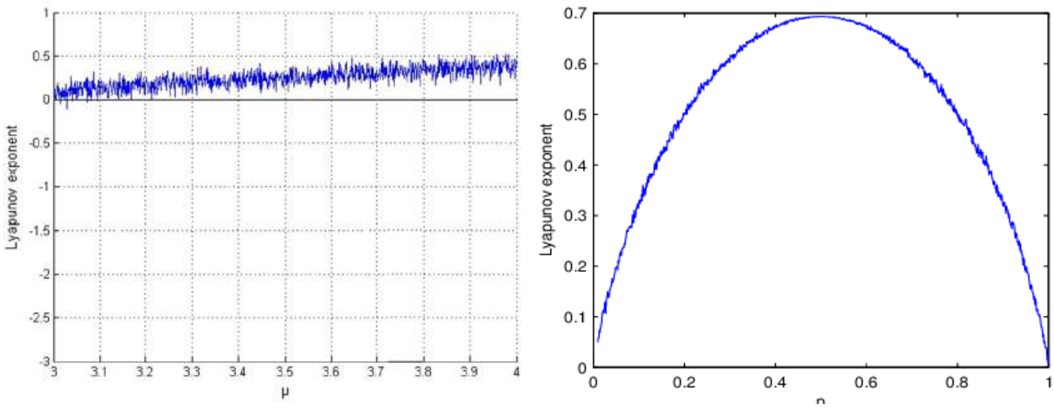
It can be seen from Figure 1, that the Lyapunov exponents of the intertwining logistic map and improved piecewise linear chaotic map are all positive which is symptomatic of the better chaotic behavior of the map.

B. ROWS AND COLUMNS' SWAPPING OPERATIONS

In the proposed scheme, images are put over one another in a 3D fashion. From these 3D images, two images are selected randomly and two randomly selected rows from these two images are swapped with each other. The same process is applied over the columns as well to achieve the confusion effects. Figure 2 shows the swapping mechanisms (row-wise) between the two 6×6 images. Figures 2(a) and 2(b) show the two random images with their pixel intensity values. Figures 2(c) and 2(d) are the updated images. The row number 2 of Figure 2(a) is swapped with row number 4 of Figure 2(b). The same operation of swapping for column number 3 of the first image and the column number 5 of the second image has been performed in Figure 3.

III. ENCRYPTION SCHEME

These multiple images encryption scheme has been proposed to encrypt N grayscale images all with the same dimensions



(a) (b)

FIGURE 1. **Lyapunov Exponent diagram of: (a) Intertwining logistic map; (b) Improved piecewise linear chaotic map.**

5	1	2	8	4	2
7	5	3	3	7	2
7	1	4	8	8	5
8	7	8	5	5	7
1	7	7	8	6	1
2	4	6	3	9	3

(a)

4	5	7	2	1	1
6	5	4	4	9	5
1	8	9	1	9	8
1	4	3	9	1	6
1	2	4	2	6	4
5	8	9	4	1	6

(b)

5	1	2	8	4	2
1	4	3	9	1	6
7	1	4	8	8	5
8	7	8	5	5	7
1	7	7	8	6	1
2	4	6	3	9	3

(c)

4	5	7	2	1	1
6	5	4	4	9	5
1	8	9	1	9	8
7	5	3	3	7	2
1	2	4	2	6	4
5	8	9	4	1	6

(d)

FIGURE 2. **Swap operation row-wise: (a) The initial state of the first image; (b) The initial state of the second image; (c) The state of the first image after swapping row number 2 with row number 4 of the second image; (d) The state of the second image after swapping row number 4 with row number 2 of the first image.**

6	3	1	5	5	8
9	7	3	2	5	3
2	3	2	9	8	8
3	2	5	5	8	2
9	8	9	5	1	3
3	8	1	4	9	5

(a)

8	7	1	4	2	1
4	9	1	4	9	2
3	5	3	1	8	9
9	5	6	4	4	3
5	1	6	3	5	8
2	4	3	2	8	5

(b)

6	3	2	5	5	8
9	7	9	2	5	3
2	3	8	9	8	8
3	2	4	5	8	2
9	8	5	5	1	3
3	8	8	4	9	5

(c)

8	7	1	4	1	1
4	9	1	4	3	2
3	5	3	1	2	9
9	5	6	4	5	3
5	1	6	3	9	8
2	4	3	2	1	5

(d)

FIGURE 3. **Swap operation column-wise: (a) The initial state of the first image; (b) The initial state of the second image; (c) The state of the first image after swapping column number 3 with column number 5 of the second image; (d) The state of the second image after swapping column number 5 with column number 3 of the first image.**

of $(L \times W)$. L and W are not necessarily equal, but they must be the same for all the chosen N images. Figure 4 demonstrates the proposed encryption scheme.

The proposed scheme consists of five phases. In Phase 1, the N number of grayscale images are stacked to make a 3D image. In Phase 2, the hash function SHA-256 has been used to get the hash codes which will in turn create the plaintext sensitivity in the proposed scheme. The SHA-256 generates a 256-bits unique value. Another 256-bits value is also provided by the user and an XOR operation has been performed between these two 256-bit values to obtain a unique 256-bits value. The incorporation of hash codes embedded the plaintext sensitivity in the proposed cipher whereas the user key enlarged the key space. The plaintext sensitivity means that a very minute change in the key will have a great effect on the obtained cipher image. Thus, it will be difficult for the potential attacker to crypt-analysis of the proposed scheme. The high key space is required to resist any potential brute force attack. The aforementioned 256-bits numbers are converted into 32 decimal values and are reshaped to get 4×8 matrices. The corresponding values of each row are added together to get 1×8 matrices. In Phase 3, the initial values and the system parameters of the chaotic maps, *i.e.*, ILM & MPWLCM are

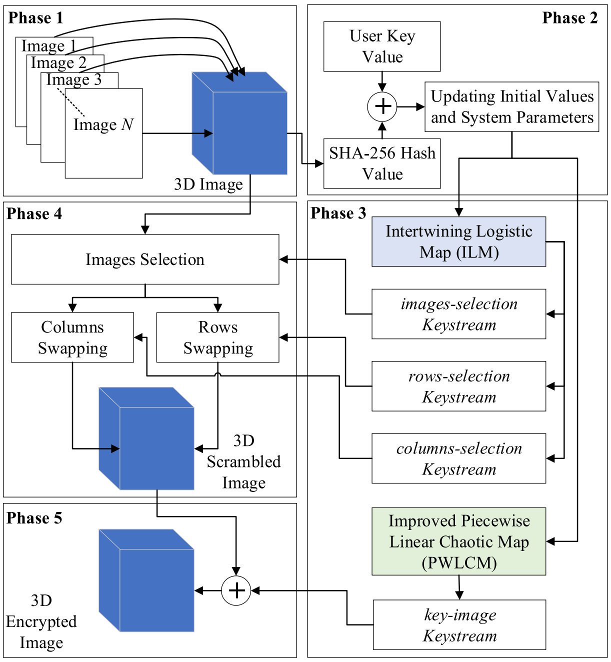


FIGURE 4. **Proposed Encryption-Scheme.**

updated from the values of the 1×8 matrices. Phase 4 is the most important phase of our work as it performs the bulk of the work, *i.e.*, scrambling/permutation of the pixels of the input images. This phase uses three key streams of *images-selection*, *rows-selection,* and *columns-selection*. The basic modus operandi of the proposed scheme works as follows. As has already been described that the given input grayscale images of the same size are stacked to form a 3D image. The *images-selection* stream selects two images randomly from the piles of images. The *rows-selection* stream selects two rows randomly from these two selected images and swaps them. Further, the third-stream *columns-selection*, as the name implies, selects two columns randomly from the selected two images and swaps these two columns. This process of selection of images, rows, and columns from the pile of images and swapping the rows and columns is iterated *LWN* times to get a 3D scrambled image. The fifth and last Phase 5 performs the diffusion or substitution of the pixels of the scrambled 3D image. An XOR operation has been carried out between the reported 3D scrambled image and the 3D *key-image* keystream generated by the MPWLCM to get the final encrypted 3D image.

A. SYSTEM PARAMETERS AND INITIAL VALUES UPDATING

Plaintext sensitivity is one of the essential features of any encryption scheme. This means that for every different input, it will produce a very different and unique cipher. Even for a single bit changed input, the scheme will produce a different and unique cipher. In the proposed scheme, the plaintext sensitivity is obtained using the hash function, i.e., SHA-256 hash value and 256-bit user key value. These two values are then XORed with each other to obtain a unique 256-bit value or 32 decimal value. The hash value is the fingerprint of the input data. The SHA-256 will produce a unique 256-bit output stream for every data. Even a single bit change in the input will produce a very different 32 decimal value. The SHA-256 hash value will be obtained from the input 3D-image in the second phase. Along with this value a user key value of equal length, i.e., 256-bits value will be provided by the user. The hash value H and user value U can be stated as follows:

$$H = h_1, h_2, \ldots, h_{32} \tag{3}$$

According to $h_i = \{h_{i,0}, h_{i,1}, \ldots, h_{i,7}\}$, wherein $h_{i,j}$, i shows the initial character value and j shows the bit number in $h_{i,j}$, In the same way, a 256-bit user key U is also divided into 8-bit blocks which can be stated as follows:

$$U = u_1, u_2, \ldots, u_{32} \tag{4}$$

According to $u_i = \{u_{i,0}, u_{i,1}, \ldots, u_{i,7}\}$, wherein $u_{i,j}$, i shows the initial character value and j shows the bit number in $u_{i,j}$, The following steps produced the initial values for ILM and MPWLCM key streams.

Step 1: Reshape each H and U into 4×8 matrices.

Step 2: Take XOR operation between H and U to obtain an updated matrix of 256 bits as:

$$U' = H \oplus U \tag{5}$$

Step 3: By adding the values of each row for all eight columns, we obtain the following:

$$c_1 = \sum_{j=1}^{4} U'(j, 1) \tag{6}$$

$$c_2 = \sum_{j=1}^{4} U'(j, 2) \tag{7}$$

$$c_3 = \sum_{j=1}^{4} U'(j, 3) \tag{8}$$

$$c_4 = \sum_{j=1}^{4} U'(j, 4) \tag{9}$$

$$c_5 = \sum_{j=1}^{4} U'(j, 5) \tag{10}$$

$$c_6 = \sum_{j=1}^{4} U'(j, 6) \tag{11}$$

$$c_7 = \sum_{j=1}^{4} U'(j, 7) \tag{12}$$

$$c_8 = \sum_{j=1}^{4} U'(j, 8) \tag{13}$$

Step 4: The system parameters for the ILM are calculated as follows:

$$k_1 = \frac{(c_1 \bigoplus c_5)}{256} + 33.50 \tag{14}$$

$$k_2 = \frac{(c_2 \bigoplus c_6)}{256} + 37.97 \tag{15}$$

$$k_3 = \frac{(c_3 \bigoplus c_7)}{256} + 35.7 \tag{16}$$

$$\mu = mod\left(\left(\frac{(c_4 \bigoplus c_8)/256}{3}\right), 3.99\right) \tag{17}$$

Step 5: The ILM initial values have been found by using the calculated system's parameters in step 4 as follows:

$$x_0 = mod\left((k_1 \times k_2 \times k_3 \times \mu), 0.5\right) \tag{18}$$

$$y_0 = mod\left(\left(\left(\frac{x_0 \times k_2 \times k_3}{\mu}\right) + k_1\right), 0.5\right) \tag{19}$$

$$z_0 = mod\left(\left(\frac{x_0 \times k_3 \times \mu \times k_1}{y_0 \times k_2}\right), 2.5\right) \tag{20}$$

where mod (x, y) gives the remainder when x is divided by y.

Step 6: In the same way, the MPWLCM initial values have been found (using the step 4 calculation of systems parameters) as follows:

$$b_0 = mod\left(\left(\frac{x_0 \times (y_0 + \mu) \times (z_0 + k_1) \times k_2}{k_2 \times 256}\right), 0.2\right) \tag{21}$$

$$n_0 = mod\left(\left(\frac{y_0 \times b_0 \times k_1 \times k_2 \times z_0}{256}\right), 0.3\right) \tag{22}$$

Step 7: By repeating the ILM chaotic system (1) for $(LWN + n_0)$ times, three chaotic sequences were generated $u = [u_1, u_2, \ldots, u_{LWN+n0}]$, $v = [v_1, v_2, \ldots, v_{LWN+n0}]$, $w = [w_1, w_2, \ldots, w_{LWN+n0}]$, where (L, W) is the dimension of a single input image, and N is the total number of images from which the 3D image is generated. Here $n_0 \geq 500$ and it is part of secret keys. To remove the transient effect of the chaotic system, we obtain the sequences by discarding the first n_0 values.

Step 8: The obtained chaotic-sequences, *i.e.*, *u*, *v*, and *w* are further processed by using the following equations to get the three keystreams. For each keystream, the two values are used simultaneously; one starting from the first position to the last position, the other from last to the first.

$$\begin{aligned} &images - selection\,(i) \\ &\qquad = mod\left(floor\left(u\,(i) \times 10^{14}\right), N\right) + 1 \end{aligned} \tag{23}$$

$$\begin{aligned} &rows - selection\,(i) \\ &\qquad = mod\left(floor\left(v\,(i) \times 10^{14}\right), L\right) + 1 \end{aligned} \tag{24}$$

$$\begin{aligned} &columns - selection\,(i) \\ &\qquad = mod\left(floor\left(w\,(i) \times 10^{14}\right), W\right) + 1 \end{aligned} \tag{25}$$

where $u(i)$, $v(i)$, and $w(i)$ are corresponding-elements of *u*, *v*, and *w* respectively. Further *images-selection*(i), *rows-selection*(i) and *columns-selection*(i) are the i^{th} elements of *images-selection*, *rows-selection* and *columns-selection* respectively. $i = 1, 2, \ldots, LWN$. Here N is the number of images and (L, W) is the dimension of the images.

Step 9: By repeating the MPWLCM chaotic system (2) for $(LWN + n_0)$ times, the single chaotic sequence was generated $b = [b_1, b_2, \ldots, b_{LWP+n0}]$. Again n_0 values are discarded to remove the transient effect.

Step 10: The obtained chaotic-sequence, *i.e.*, *b* is further processed by using the following equation to get a single keystream.

$$key - image\,(ki) = mod\left(floor\left(b\,(ki) \times 10^{14}\right), 256\right), \tag{26}$$

where $b(ki)$ and *key-image*(ki) are the ki^{th} elements of *b* and *key-image* respectively. $ki = 1, 2, \ldots, LWN$.

B. ENCRYPTION PROCEDURE

The proposed scheme encrypts the N grayscale images in a 3D image space. For each image in the 3D image, the size is $L \times W$. The following steps provide the detail of the encryption procedure.

Step 1: (Building 3D image): N grayscale same sized images are put over one another in such a fashion that a 3D large image *3D-image* (say) is formed.

Step 2: (Permutation operation):

Step 2.1: Initialize *index*1=1

Step 2.2: Initialize *index*2=*LWN*

Step 2.3: Select two images from the *3D-image*, using *index*1 and *index*2.

$$im1 = images - selection\,(index1) \tag{27}$$

$$im2 = images - selection\,(index2) \tag{28}$$

Step 2.4: Select two rows, one from *im*1 and one from *im*2, using *index*1 and *index*2

$$r1 = rows - selection\,(index1) \tag{29}$$

$$r2 = rows - selection\,(index2) \tag{30}$$

Step 2.5: Select two columns, one from *im1* and one from *im2*, using *index1* and *index2*

$$c1 = columns - selection\,(index1) \tag{31}$$

$$c2 = columns - selection\,(index2) \tag{32}$$

Step 2.6: (Swapping/scrambling operation): The following steps carry out the swapping operation for the selected images, rows, and columns. A temporary memory location *temp* has been used. Equations (33) to (35) are for swapping of rows and equations (36) to (38) for columns.

$$temp = 3D - image\,(r1, :, im1) \tag{33}$$

$$3D - image\,(r1, :, im1) = 3D - image(r2, :, im2) \tag{34}$$

$$3D - image(r2, :, im2) = temp \tag{35}$$

$$temp = 3D - image\,(:, c1, im1) \tag{36}$$

$$3D - image\,(:, c1, im1) = 3D - image(:, c2, im2) \tag{37}$$

$$3D - image(:, c2, im2) = temp \tag{38}$$

Step 2.7: index $1 = index\ 1 + 1$

Step 2.8: index $2 = index\ 2 - 1$

Step 2.9: Repeat steps 2.3 to step 2.9 while $index1 \leq LWN$ and $index2 \geq 1$ to obtain the scrambled 3D image $3D - image'$ (say).

Step 3: (Substitution operation): Reshape $key - image$ to $L\times\ W\ \times N$ and take the XOR operation between the $3D - image'$ and $key - image$.

$$3D - image'' = 3D - image' \oplus key - image \tag{39}$$

The $3D - image''$ is the final encrypted image.

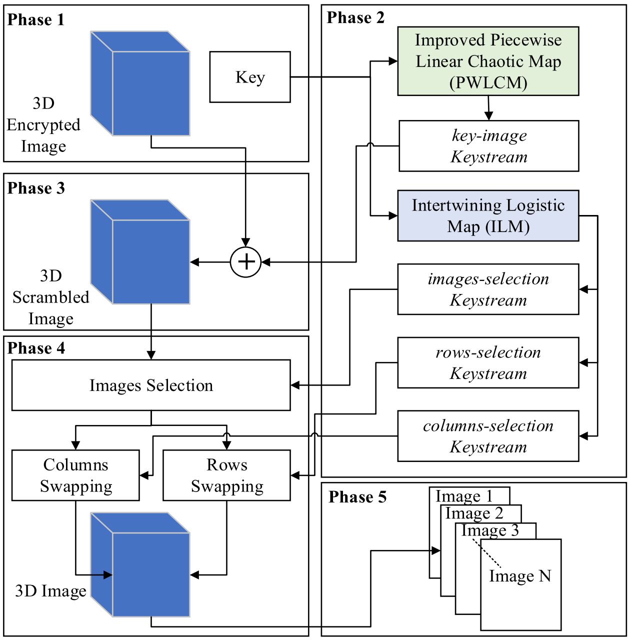


FIGURE 5. **Proposed Decryption Scheme.**

C. DECRYPTION PROCEDURE

Since the proposed multiple images encryption scheme is symmetric so its decryption procedure will consist of the exact reverse steps of the encryption procedure. The decryption procedure has been shown in Figure 5.

The following steps describe the decryption procedure in detail.

Step 1: Input the 3D encrypted-image $3D - image''$ and key.

Step 2: Generate the key streams $images - selection$, $rows - selection$, $columns - selection$, and $key - image$, as described in section 3.1.

Step 3: Nullifying the diffusion effects using $key - image$ keystream by performing the following XOR operation.

$$3D - image' = 3D - image'' \oplus key - image \tag{40}$$

$3D - image'$ is just the scrambled image with no diffusion effects.

Step 4: (Permutation operation):

Step 4.1: Initialize $index\ 1 = 1$

Step 4.2: Initialize $index\ 2 = LWN$

Step 4.3: Select two images from the 3D image $3D - image'$, using $index1$ and $index2$.

$$im1 = images - selection\,(index1) \tag{41}$$

$$im2 = images - selection\,(index2) \tag{42}$$

Step 4.4: Select two rows, one from $im1$ and one from $im2$, using $index1$ and $index2$

$$r1 = rows - selection\,(index1) \tag{43}$$

$$r2 = rows - selection\,(index2) \tag{44}$$

Step 4.5: Select two columns, one from $im1$ and one from $im2$, using $index1$ and $index2$

$$c1 = columns - selection\,(index1) \tag{45}$$

$$c2 = columns - selection\,(index2) \tag{46}$$

Step 4.6: Swapping operation: A temporary memory location $temp$ is used to store the rows or columns values during swapping operation.

$$temp = 3D - image\,(r1, :, im1) \tag{47}$$

$$3D - image\,(r1, :, im1) = 3D - image(r2, :, im2) \tag{48}$$

$$3D - image(r2, :, im2) = temp \tag{49}$$

$$temp = 3D - image\,(:, c1, im1) \tag{50}$$

$$3D - image\,(:, c1, im1) = 3D - image(:, c2, im2) \tag{51}$$

$$3D - image(:, c2, im2) = temp \tag{52}$$

Step 4.7: $index1 = index1 + 1$

Step 4.8: $index2 = index2 - 1$

Step 4.9: Repeat steps 4.3 to step 4.9 while $index1 \leq LWN$ and $index2 \geq 1$ to finally get the unscrambled 3D image or in other words, the original plain images. Separate each layer as a separate image and save them to obtain the N plain images.

IV. SIMULATION

In the realm of images cryptography, a plethora of attacks (brute-force attack, differential attack, statistical attack, histogram attack, chosen-plaintext attack, entropy attack, etc.) exists. Any good image cipher is expected to endure these attacks from potential hackers and intruders. To demonstrate the effectiveness and practical utility of the proposed multiple grayscale images encryption schemes, 9 grayscale plain images have been selected. The names of these images are Lena, Baboon, Barbara, Camera man, Airplane, Bridge, Chemical plant, Clock and Couple with four different sizes, *i.e.*, 128×128, 256×256, 512×512 and 1024×1024. From these images, six 3D images were developed, *i.e.*, $4 \times 128 \times 128$, $9 \times 128 \times 128$, $4 \times 256 \times 256$, $4 \times 512 \times 512$, $9 \times 512 \times 512$ and $9 \times 1024 \times 1024$ to show the capability of the proposed cipher for varied sizes. The 2D counterparts of these images are 256×256, 384×384, 512×512, 1024×1024, 1536×1536 and 3072×3072 respectively. The above-mentioned images can be accessed from (http://www.vision.caltech.edu/visipedia/CUB-200-2011.html).

The initial values and the systems parameters taken for ILM and MPWLCM are: $x_0 = 0$, $y_0 = 0$, $z_0 = 0$, $\mu = 0$, $k_1 = 33.5$, $k_2 = 37.9$, $k_3 = 35.7$, $b_0 = 0$, $n_0 = 0$. Further, the four user keys provided to the simulation are: uk_1 = 'aa11bb22', uk_2 = 'aa11bb22', uk_3 = 'aa11bb22', uk_4 = 'aa11bb22'. Figure 6 shows the nine original grayscale images. Figure 7(a) is a combined large image formed from Lena, Baboon, Barbara, and Camera man images whereas Figures 7(b) to 7(d) show the scrambled large

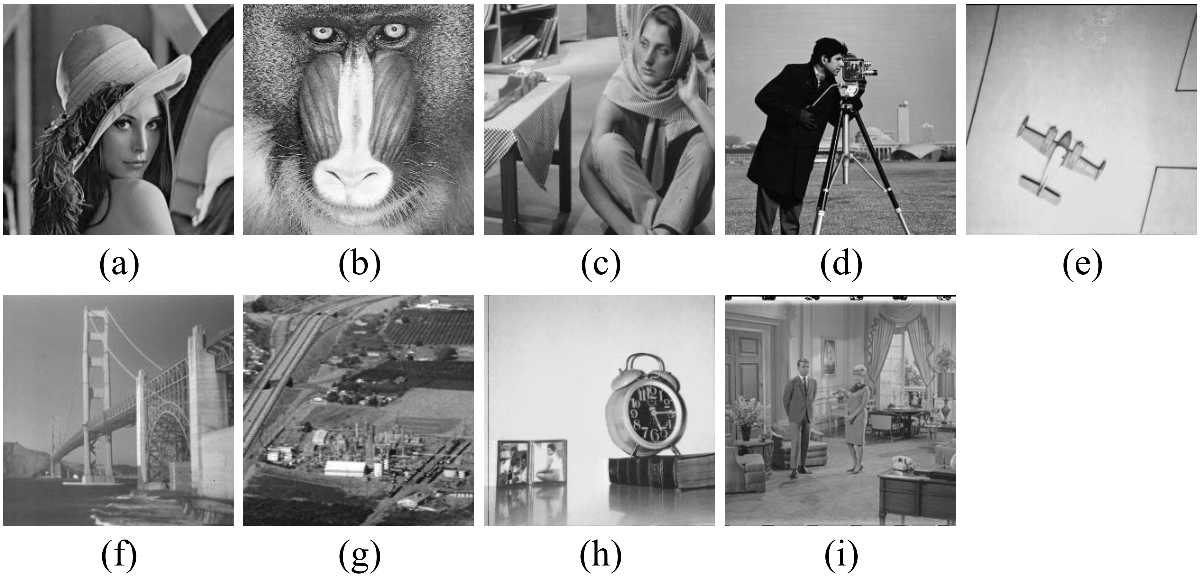


FIGURE 6. The original images: (a) Lena-plain-image; (b) Baboon-plain-image; (c) Barbara-plain-image; (d) Camera-man-plain-image; (e) Air-plane-plain-image; (f) Bridge-plain-image; (g) Chemical-plant-plain-image; (h) Clock-plain-image; (i) Couple-plain-image.

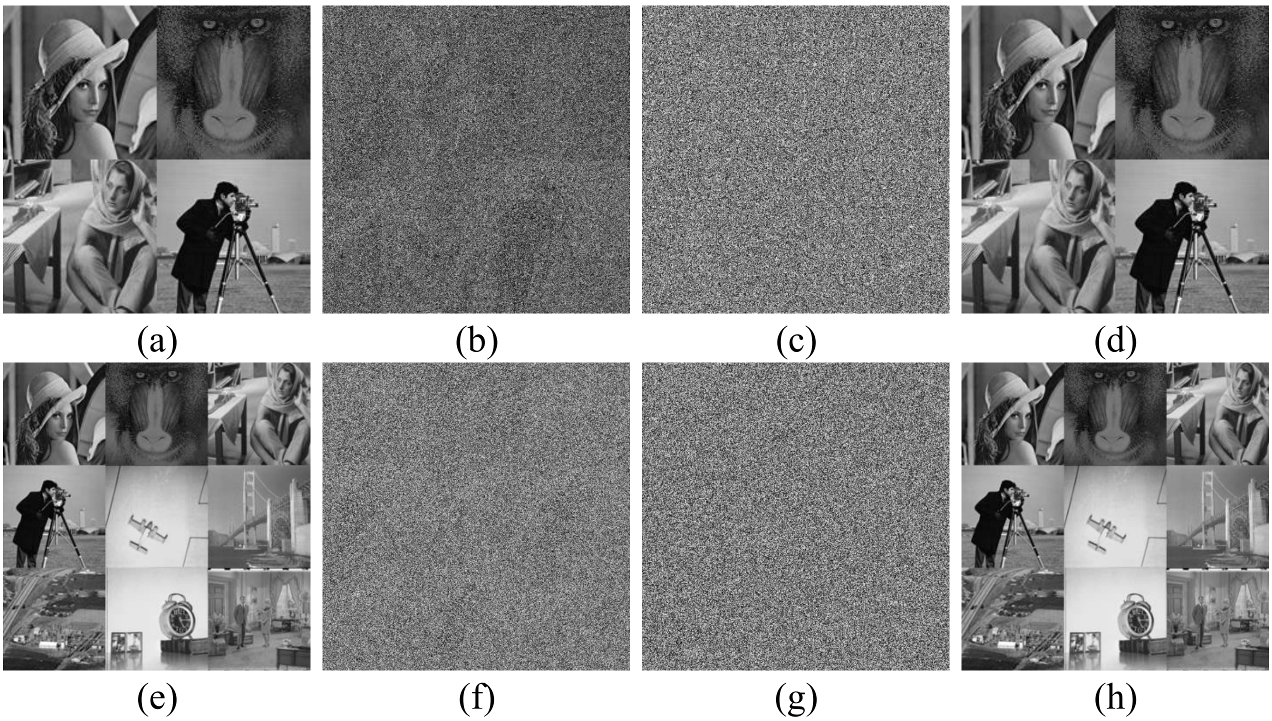


FIGURE 7. (a) Combined 4 images; (b) Scrambled image of (a); (c) Encrypted image of (a); (d) Decrypted image of (a); (e) Combined 9 images; (f) Scrambled image of (e); (g) Encrypted image of (e); (h) Decrypted image of (e).

image, encrypted large image, and decrypted large image respectively. The same treatment has been carried out in Figures 7(e) to 7(h) for the nine images Lena, Baboon, Barbara, Camera man, Airplane, Bridge, Chemical plant, Clock, and Couple. These figures depict the success of the proposed MIE since the encrypted images have been completely turned into a format, which is not recognizable. Since the images taken with different sizes of $4 \times 128 \times 128$, $9 \times 128 \times 128$, $4 \times 256 \times 256$, $4 \times 512 \times 512$, $9 \times 512 \times 512$, $9 \times 1024 \times 1024$ will be frequently referred, so for the sake of brevity they will be called as *3D-1*, *3D-2*, *3D-3*, *3D-4*, *3D-5* and *3D-6* respectively.

V. SECURITY ANALYSIS

In this section, performance analyses using the different validation metrics will be carried out.

A. KEY SPACE ANALYSIS

A brute force attack is crafted by the attackers to systematically exhaust all the possible keys on a scheme until the secret key is not found. The larger the key space, the more time will it take. The proposed scheme consists of 256-bit hash values and a 256-bit user key. The hash value is obtained from the input image, while the user key is provided by the user which is split into four subparts called user keys, *i.e.*, uk_1, uk_2, uk_3, and uk_4. Each user key consisting of eight bytes or sixteen hexadecimal numbers, which make 256-bits or 32-bytes or 64 hexadecimal numbers. The user key values used in the simulation are [ff, db, 02, 27, 67, 96, 07, d6, 4b, 8b, e4, 38, 4d, 4e, 13, 26, 33, 7c, 39, 4b, 92, da, 81, e4, 96, 01, 87, b5, d4, 9c, c5, f9] It contributes $(2^8)^{32} = 2^{256}$ to the key space. Apart from this, nine variables (three initial values and the four system parameters) of the ILM and (one initial value and one system parameter) of the MPWLCM contribute $(10^{14})^9 = 10^{126}$ to the key space if computer precision of 10^{-14} is taken. So, the total key space of the proposed cipher comes out to be $2^{256} \times 10^{126} \approx 1.157 \times 10^{203}$. This key-space is more than enough to resist the brute-force-attack. It also fulfills the minimum requirement of 2^{100} [28], [40]. Moreover, Table 1 compares the key space between the proposed cipher and other published works.

TABLE 1. Key space comparison with some other schemes.

Algorithm	Key Space
Ours	1.157×10^{203}
Ref. [25]	1.936×10^{59}
Ref. [26]	3.402×10^{128}
Ref. [27]	10^{220}
Ref. [28]	7.370×10^{134}
Ref. [30]	5.841×10^{135}
Ref. [31]	10^{60}
Ref. [41]	4.34×10^{96}
Ref. [42]	10^{75}
Ref. [43]	10^{56}
Ref. [44]	10^{60}

B. KEY SENSITIVITY ANALYSIS

The key sensitivity is one of the most important features of any encryption scheme. It means that for any encryption process, the verbatim key will be used for the decryption. A very slight change in the key will have a very great effect on the output. The key sensitivity is tested in two different methods. In the first method, an image is encrypted using some key. The same image is encrypted with another key that is very minutely different from the first key, *i.e.*, 10^{-14}, these both encrypted images will be entirely different from each other. In the second method, a failed attempt is made to decrypt the encrypted image with a slightly different key. Again, the encrypted image is not decrypted successfully unless the correct key is not employed.

For the first method, 3D image is encrypted using two key sets, say, K_0 and K_1 with very minute difference [45]. K_0 is the straightforward key set formed by the initial values and the system parameters of the chaotic systems being used, *i.e.*, $K_0 = \{x_0, y_0, z_0, \mu, k_1, k_2, k_3, b_0, n_0\}$. The 3D image of Figure 8(a) has been encrypted by K_0 and the resultant encrypted image has been shown in Figure 8(b). To incorporate the key sensitivity, add a very minute value of $\Delta = 10^{-14}$ in x_0 to obtain x_0' i.e. $x'_0 = x_0 + \Delta$ Let the new key set formed is $K_1 = \{k'_1, k_2, k_3, \mu, x_0, y_0, z_0, b_0, n_0\}$ key set. Now K_1 has been used to encrypt the same image of Figure 8(a) and output has been shown in Figure 8(c). Figure 8(b) and Figure 8(c) are two different encrypted images of the same input image Figure 8(a). Figure 8(d) shows the differential image of these two encrypted images, *i.e.*, Figure 8(b) and Figure 8(c). This differential image has been obtained by taking the absolute value of the difference between the corresponding pixel intensity values of the encrypted images. Furthermore, these two images are 99.0209% different from each other as far as the pixel intensity values are concerned. This shows the implication of a very minute change in the key. Apart from that, Figure 8(e) is the decrypted image of Figure 8(b) using the correct key K_0. Further, Figure 8(f) shows the decrypted image of Figure 8(b) using the incorrect key K_1. In the same manner, Figure 8(g) shows the decrypted image of Figure 8(c) using correct key K_1. Lastly, Figure 8(h) shows the decrypted image of Figure 8(c) using incorrect key K_0.

Table 2 shows the key sensitivity in a more elaborate way. This table demonstrates the difference rate of the pixels between the two encrypted images obtained via K_0, and K_a ($a = 1, 2, \ldots, 18$). Table 2 shows that the minimum difference rate between the two encrypted images is 98.7122%. Further, 99.6223% is the maximum and 99.0924% is the average value which is better than [2], [46], [47]. Therefore, the proposed algorithm is better.

C. STATISTICAL ANALYSIS

Statistical analysis is yet another metric to gauge the efficiency of any encryption scheme. Usually, it encompasses two tests, *i.e.*, histogram analysis and correlation coefficient analysis.

1) HISTOGRAM ANALYSIS

A histogram of an image shows the distribution of its pixels' intensity values. The plain image pixels are fashioned in a peculiar way due to which the histogram made through it has a typical slanting bar which is full of information. This slanting bar is open to the histogram attack. To resist such an attack, the encryption algorithm should be such that it should render an encrypted image whose histogram should have a uniform bar. Of course, this uniform bar is safe from a histogram attack. Figures 9(a) and 9(b) show the plain image and encrypted image histograms respectively. One can see that the histogram made through the encrypted image has

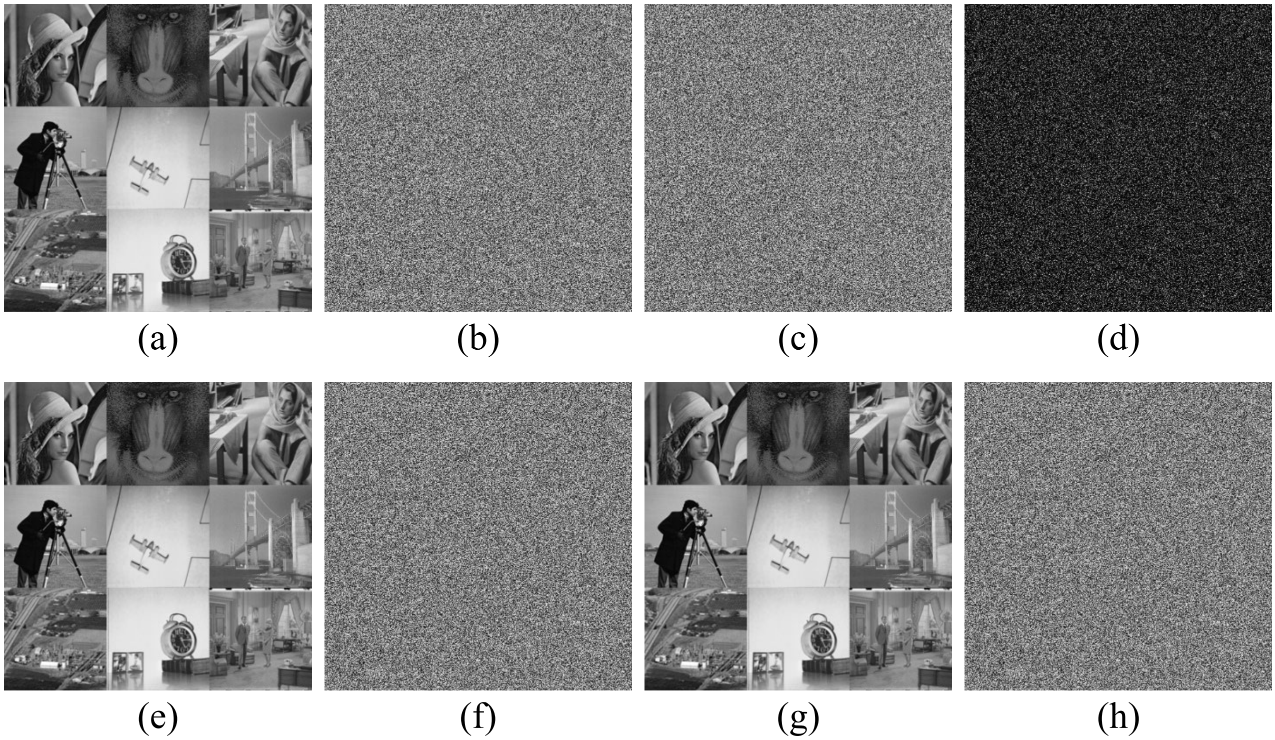


FIGURE 8. Keys-Sensitivity-Assessment: (a) combined image; (b) Encrypted Image of (a) with K0; (c) Encrypted Image of (a) with K1; (d) The differential image between (b) and-(c); (e) Decrypted Image from (b) with exact key set K0; (f) Decrypted Image from (b) with incorrect-key set K1; (g) Decrypted-Image from (c) with the-exact key set K1 ; (h). Decrypted Image-from (c).with incorrect key set K0.

TABLE 2. Difference rate between two encrypted images by minutely changed keys.

Secret keys	Difference rates (%)					
	3D-1	*3D-2*	*3D-3*	*3D-4*	*3D-5*	*3D-6*
K_1 ($x'_0 = x_0 + \Delta$)	98.6459	99.0221	98.8524	98.8772	99.0355	99.0306
K_2 ($x'_0 = x_0$ - Δ)	98.7460	99.0126	98.9698	98.9672	99.0205	99.0257
K_3 ($y'_0 = y_0 + \Delta$)	98.8370	99.0173	98.8765	98.7122	99.0206	99.0248
K_4 ($y'_0 = y_0$ - Δ)	98.9849	99.0160	98.9768	98.9484	99.0283	99.0287
K_5 ($z'_0 = z_0 + \Delta$)	98.7139	99.0085	98.9719	98.9616	99.0292	99.0272
K_6 ($z'_0 = z_0$ - Δ)	98.9620	99.0255	98.9020	98.9565	99.0287	99.0269
K_7 ($y'_0 = y_0 + \Delta$)	99.6155	99.6162	99.6223	99.6161	99.6096	99.6086
K_8 ($y'_0 = y_0$ - Δ)	99.6109	99.6202	99.6197	99.6108	99.6115	99.6091
K_9 ($z'_0 = z_0 + \Delta$)	99.5728	99.6223	99.6181	99.6255	99.6151	99.6133
K_{10} ($z'_0 = z_0$ - Δ)	99.6017	99.5911	99.5987	99.6108	99.6118	99.6108
K_{11} ($k'_1 = k_1 + \Delta$)	98.9864	99.0072	98.8307	98.9483	99.0287	99.0209
K_{12} ($k'_1 = k_1$ - Δ)	98.8261	99.0214	98.7848	98.7558	99.0354	99.0317
K_{13} ($k'_2 = k_2 + \Delta$)	98.9597	99.0044	98.7561	98.8171	99.0251	99.0271
K_{14} ($k'_2 = k_2$ - Δ)	98.9139	99.0411	98.7499	98.8663	99.0425	99.0306
K_{15} ($k'_3 = k_3 + \Delta$)	98.9246	99.0031	98.7074	98.9179	99.0204	99.0246
K_{16} ($k'_3 = k_3$ - Δ)	98.7307	98.9936	99.1886	98.9514	99.0201	99.0213
K_{17} ($\mu' = \mu + \Delta$)	98.8566	98.9705	99.3413	98.9675	99.0273	99.0283
K_{18} ($\mu' = \mu - \Delta$)	98.9398	98.9651	98.9463	98.9522	99.0197	99.0209
Average	**99.0238**	**99.1421**	**99.0157**	**99.0590**	**99.1572**	**99.1562**
Average of All	**99.0924**					

a uniform bar over it, which is symptomatic to the fact that the pixels' intensity values have been uniformly distributed in the entire image.

Also, variance values showing the uniformity of the histograms of the cipher images have been calculated (Table 3). The closer the variance value to 5400, the higher the

TABLE 3. The variance values of histograms of the cipher-images using different keys.

Keys	*3D-1*	*3D-2*	*3D-3*	*3D-4*	*3D-5*	*3D-6*	**Average**
K_0	5459.4952	5463.0304	5473.0535	5462.8041	5462.7906	5463.9108	**5464.1808**
K_1	5462.4749	5477.8209	5465.5305	5455.5849	5457.9106	5461.3096	**5463.4386**
K_2	5470.1297	5464.1000	5471.4717	5460.2285	5462.9910	5460.4375	**5464.8931**
K_3	5453.9173	5469.2764	5473.7083	5468.8828	5463.5047	5460.1402	**5464.9050**
K_4	5441.2406	5442.3993	5459.1582	5465.1352	5466.0113	5458.5692	**5455.4190**
K_5	5460.5371	5476.1462	5472.4532	5464.7794	5464.2456	5460.1200	**5466.3803**
K_6	5439.4427	5464.0181	5454.9369	5465.2885	5460.4905	5459.9142	**5457.3485**
K_7	5480.3382	5464.7537	5464.7495	5462.3319	5466.5928	5460.2019	**5466.4947**
K_8	5476.8101	5451.4416	5453.8912	5457.5080	5457.9922	5461.2176	**5459.8101**
K_9	5483.7882	5444.9342	5459.9134	5457.8391	5463.3275	5461.1559	**5461.8264**
K_{10}	5449.7688	5455.8564	5462.5661	5457.4634	5464.2718	5462.7069	**5458.7722**
K_{11}	5483.5981	5457.7801	5467.3043	5452.4360	5466.4231	5460.3289	**5464.6451**
K_{12}	5470.9903	5458.1584	5467.9434	5467.1521	5463.8676	5461.0112	**5464.8538**
K_{13}	5451.2155	5450.8884	5479.2748	5462.7757	5465.3274	5459.8604	**5461.5570**
K_{14}	5452.5305	5458.8374	5451.4088	5461.5084	5461.4957	5459.0663	**5457.4745**
K_{15}	5438.7179	5439.3962	5462.9198	5460.8273	5459.2589	5461.1223	**5453.7071**
K_{16}	5435.3193	5458.1782	5457.8733	5457.8359	5460.6943	5464.3030	**5455.7007**
K_{17}	5467.0147	5449.9590	5462.1019	5467.6623	5460.9425	5459.2166	**5461.1495**
K_{18}	5472.1213	5449.1423	5464.2431	5462.1130	5456.5338	5460.3289	**5460.7471**
Average of All							**5461.2265**

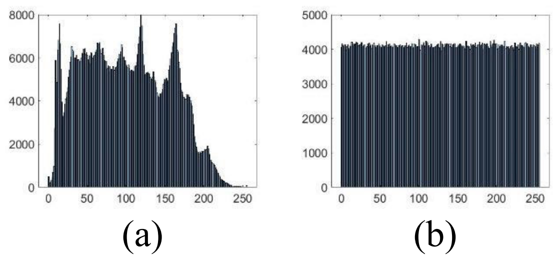


FIGURE 9. Histogram analysis: (a) Histogram of a plain image; (b) Histogram of an encrypted image.

uniformity [45]. The first row of Table 3 shows the variance results of six images, *i.e*., *3D-1*, *3D-2*, *3D-3*, *3D-4*, *3D-5*, *3D-6* against the key K_0 while the results of the remaining rows have been obtained from the K_a ($a = 1, 2, \ldots, 18$) as discussed in section 5.2. The average of the variance values is 5461.2265 which is less than those in [26], [27], [30]. Apart from this, the variance value of the histogram of the original images are about 22525, 132885, 440146, 4536850, 15425113 and 132720434 for *3D-1, 3D-2, 3D-3, 3D-4, 3D-5, 3D-6* respectively. This once again depicts the better security effects.

Furthermore, the influence of altering the secret keys on the uniformity of the histograms of the cipher images has been investigated. For this purpose, the variance difference percentages between the two cipher images have been calculated. The first cipher image was obtained using the key K_0 and the second through the key K_a($a = 1, 2, \ldots, 18$). The results are shown in the Table 4. The average value is 0.1312%. Further, the highest percentage variance value is 0.4450%, being less than [46], [48], [49], and the lowest value is 0.0072% which is again less than [45], [49], [50]. The results once again demonstrate the better security effects.

2) CORRELATION COEFFICIENT ANALYSIS

The adjacent pixels of any plain image are highly correlated. Any two pixels sharing their boundaries are called the adjacent pixels. They can be vertically, horizontally, or diagonally adjacent. A metric called correlation coefficient is calculated to determine the correlation of the adjacent pixels. The basic aim of any image encryption scheme is to break this correlation among the adjacent pixels. To determine this coefficient, we randomly chose 8,000 pairs of adjacent pixels in the three dimensions, *i.e*., vertical, horizontal, and diagonal. The related mathematical formula for the coefficient is given below [51]:

$$CC = \frac{N\sum_{j=1}^{N}\left(a_j \times b_j\right) - \sum_{j=1}^{N} a_j \times \sum_{j=1}^{N} b_j}{\sqrt{(N\sum_{j=1}^{N} a_j^2 - (\sum_{j=1}^{N} a_j^2))(N\sum_{j=1}^{N} b_j^2 - (\sum_{j=1}^{N} b_j^2))}} \tag{53}$$

In equation (53), a and b denote the pixel intensity values for the given image, N is the total number of pixels in the image. The correlation distribution of the image *3D-5* for both the plain image and encrypted image is shown in Figure 10.

Table 5 gives the correlation coefficients of two adjacent pixels for both the original images and the encrypted

TABLE 4. Percentage of variance difference of histograms of the cipher-images.

Keys	*3D-1*	*3D-2*	*3D-3*	*3D-4*	*3D-5*	*3D-6*	**Average**
K_1	0.0546	0.2707	0.1375	0.1322	0.0893	0.0476	**0.1220**
K_2	0.1948	0.0196	0.0289	0.0471	0.0037	0.0636	**0.0596**
K_3	0.1022	0.1143	0.0120	0.1113	0.0131	0.0690	**0.0703**
K_4	0.3344	0.3776	0.2539	0.0427	0.0590	0.0978	**0.1942**
K_5	0.0191	0.2401	0.0110	0.0362	0.0266	0.0694	**0.0671**
K_6	0.3673	0.0181	0.3310	0.0455	0.0421	0.0731	**0.1462**
K_7	0.3818	0.0315	0.1517	0.0086	0.0696	0.0679	**0.1185**
K_8	0.3172	0.2121	0.3501	0.0969	0.0878	0.0493	**0.1856**
K_9	0.4450	0.3312	0.2401	0.0909	0.0098	0.0504	**0.1946**
K_{10}	0.1782	0.1313	0.1916	0.0978	0.0271	0.0220	**0.1080**
K_{11}	0.4415	0.0961	0.1050	0.1898	0.0665	0.0656	**0.1607**
K_{12}	0.2106	0.0892	0.0934	0.0796	0.0197	0.0531	**0.0909**
K_{13}	0.1517	0.2223	0.1137	0.0005	0.0464	0.0741	**0.1014**
K_{14}	0.1276	0.0768	0.3955	0.0237	0.0237	0.0887	**0.1226**
K_{15}	0.3806	0.4326	0.1852	0.0362	0.0647	0.0510	**0.1917**
K_{16}	0.4428	0.0888	0.2774	0.0909	0.0384	0.0072	**0.1576**
K_{17}	0.1377	0.2393	0.2001	0.0889	0.0338	0.0859	**0.1310**
K_{18}	0.2313	0.2542	0.1610	0.0127	0.1145	0.0656	**0.1399**
Average of All							**0.1312**

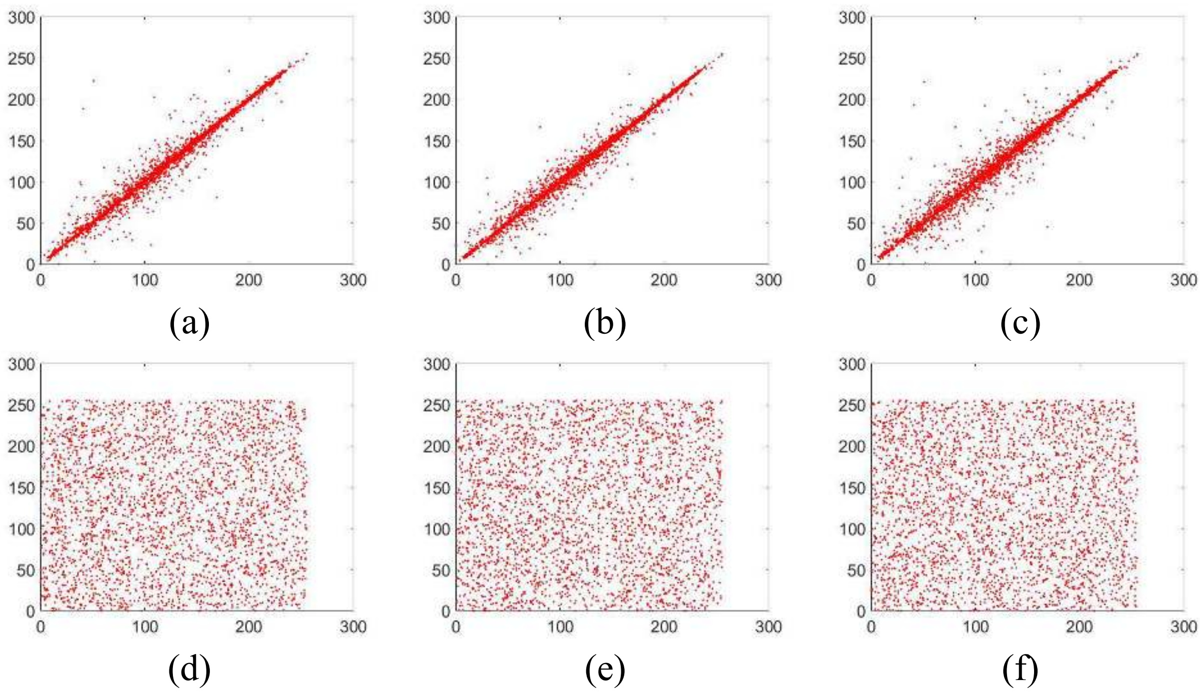


FIGURE 10. Correlation-distribution of adjacent-pixels for the image 3D-5: (a) Distribution-of horizontally adjacent-pixels of the plain image; (b) Distribution of-vertically adjacent-pixels of the plain image; (c) Distribution-of diagonally-adjacent pixels of the plain image; (d) Distribution of horizontally-adjacent pixels of the encrypted-image; (e) Distribution-of vertically adjacent pixels of the encrypted image; (f) Distribution of diagonally adjacent pixels of the encrypted image.

images of *3D-1, 3D-2, 3D-3, 3D-4, 3D-5, 3D-6*. The table indicates that the correlation coefficients between the cipher images are very close to 0, whereas it is very close to 1 for the plain images. Both Figure 10 and Table 5 jointly demonstrate that after the proposed encryption scheme gets applied over the plain images, the correlation among the plain images and encrypted images has been reduced copiously. Table 5, further, draws a comparison of the correlation coefficient between the proposed work and the other published works [25], [26], [29].

TABLE 5. Correlation coefficient for *3D-1*, *3D-2*, *3D-3*, *3D-4*, *3D-5* and *3D-6* images and its correlation direction Horizontal, Vertical and Diagonal.

Encryption Algorithm	Correlation direction					
	Image	Image Size	No. of Images	Horizontal	Vertical	Diagonal
Ours	*3D-1*	256x256	4x128x128	0.0012	0.0048	-0.0013
	3D-2	384x384	9x128x128	0.0016	0.0012	0.0045
	3D-3	512x512	4x256x256	-0.0028	-0.0018	0.0022
	3D-4	1024x1024	4x512x512	0.0023	0.0052	0.0024
	3D-5	1536x1536	9x512x512	0.0016	-0.0023	-0.0012
	3D-6	3072x3072	9x1024x1024	0.0011	-0.0048	0.0034
Ref. [25]		512x512	4x256x256	-0.1592	0.0067	-0.0575
Ref. [53]		384x384	9x128x128	0.0079	0.0052	0.0116
		1536x1536	9x512x512	0.0018	0.0041	0.0021
		3072x3072	9x1024x1024	0.0037	0.0059	0.0062
Ref. [29]		256x256	4x128x128	0.0029	0.0017	0.0009
		384x384	9x128x128	0.0051	0.0068	0.0013
		512x512	4x256x256	0.0022	0.0016	0.0007
		1024x1024	4x512x512	0.0016	0.0009	0.0007
		1536x1536	9x512x512	0.0034	0.002	0.0007
		3072x3072	9x1024x1024	0.0009	0.0015	0.0008
Ref. [26]		512x512	4x256x256	0.0073	−0.0014	−0.0043

TABLE 6. The results of information entropy analysis.

Encryption Algorithm	Image	Image Size	No. of Images	Original Image	Encrypted
Ours	*3D-1*	256x256	4x128x128	7.65599	7.99691
	3D-2	384x384	9x128x128	7.76589	7.99884
	3D-3	512x512	4x256x256	7.65283	7.99933
	3D-4	1024x1024	4x512x512	7.66602	7.99985
	3D-5	1536x1536	9x512x512	7.77178	7.99992
	3D-6	3072x3072	9x1024x1024	7.77345	7.99998
Ref. [53]		384x384	9x128x128		7.99720
		1536x1536	9x512x512		7.99880
		3072x3072	9x1024x1024		7.99900
Ref. [29]		384x384	9x128x128		7.99870
		1536x1536	9x512x512		7.99920
		3072x3072	9x1024x1024		7.99940
Ref. [26]		256x256		7.3446	7.99700
		512x512		7.1914	7.99930
		1024x1024		6.7327	7.99980
Ref. [28]		512x512			7.99930

D. INFORMATION ENTROPY ANALYSIS

Information entropy is used to measure the degree of unpredictability and randomness in a particular information source. The mathematical formula (54) was introduced by Shannon in 1949 [52]:

$$H\ (m) = \sum\nolimits_{i=0}^{2^n-1} p(m_i) log \frac{1}{p(m_i)} \tag{54}$$

In the above equation, the information entropy has been denoted by $H(m)$. Here m is the information source. $p(m_\mathrm{i})$ is the probability of m_i. 8 is the maximum value of information entropy for an image with 256 gray values. It means that the closer the resultant value of information entropy to 8 is, the more randomness will be there in the information source and tougher the prediction will be [28]. Table 6 shows the information entropies of the chosen images. The average value for the entropy of all the encrypted images is very close to 8, which shows that the proposed cipher is very much immune to the entropy attack. Table 6 also provides a comparison between the results of the proposed work and the literature. The proposed algorithm is better than [26], [28], [29], [53] as far as entropy is concerned.

E. DIFFERENTIAL ATTACK ANALYSIS

Sometimes the hackers and intruders' resort to this attack to have access over the secret key. In this particular attack, a hacker makes a very slight change in the single pixel of the input image. Now both the images, one without any change and the other with a slight change, are encrypted.

TABLE 7. Average NPCR and UACI results.

Encryption Algorithm	Image	Image Size	No. of Images	*NPCR*	*UACI*
Ours	*3D-1*	256x256	4x128x128	99.6185	33.6068
	3D-2	384x384	9x128x128	99.6195	33.3273
	3D-3	512x512	4x256x256	99.6178	33.5280
	3D-4	1024x1024	4x512x512	99.6168	33.4877
	3D-5	1536x1536	9x512x512	99.6135	33.4585
	3D-6	3072x3072	9x1024x1024	99.6103	33.4683
			Average	**99.6161**	**33.4794**
Ref. [53]		384x384	9x128x128	99.1852	32.8503
		1536x1536	9x512x512	99.2188	33.1728
		3072x3072	9x1024x1024	98.9907	33.1569
Ref. [29]		384x384	9x128x128	99.5841	33.3182
		1536x1536	9x512x512	99.5188	33.2638
		3072x3072	9x1024x1024	99.6653	33.3857
Ref. [26]		256x256		99.5837	33.4616
		512x512		99.6073	33.4067
		1024x1024		99.6104	33.4534
Ref. [28]		512x512		99.6100	33.4500

The resultant cipher images have a potential relationship between them. This potential relationship can be figured out if these cipher images are manipulated ingeniously. To cope with this attack, two measures of *NPCR* and *UACI* are normally used. The former stands for the number of pixels change rate and the latter for unified average changing intensity. Through the usage of the mathematical formulas given below in (55) and (57), one can evaluate the effect of single-pixel changing in the plain image over the encrypted image.

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\% \tag{55}$$

where M and N represent the width and height of the image respectively.

$D(i,j)$ can be defined by:

$$D(i,j) = \begin{cases} 1, & if\ C(i,j) \neq C'(i,j), \\ 0, & if\ C(i,j) \neq C'(i,j) \end{cases} \tag{56}$$

$$UACI = \frac{1}{M \times N}\left[\sum_{i,j} \frac{\left|C(i,j) - C'(i,j)\right|}{255}\right] \times 100\% \tag{57}$$

C and C' are respectively the ciphered images before and after one pixel of the plain image is changed.

The *NPCR* and *UACI* values of the chosen images must close to 100% and 33.3% respectively to cope with the differential attack. The results of the proposed scheme are shown in Table 7. The averages of *NPCR* and *UACI* of the six different images are 99.6161% and 33.4794% respectively. The results prove that the proposed encryption scheme is strong enough to cope with the differential attacks of *NPCR* and *UACI*. Further, the comparison between our calculated *NPCR* and *UACI* values with some other algorithms is also drawn in Table 7 [26], [28], [29], [53].

TABLE 8. Critical values (percentage) for NPCR randomness test.

Size	$N^*_{0.05}$	$N^*_{0.01}$	$N^*_{0.001}$
256 × 256	99.5693	99.5527	99.5341
384 × 384	99.5827	99.5716	99.5592
512 × 512	99.5893	99.5810	99.5717
1024 × 1024	99.5994	99.5952	99.5906
1536 × 1536	99.6027	99.5999	99.5968
3072 × 3072	99.6060	99.6047	99.6031

TABLE 9. Theoretical results (percentage) for UACI randomness test.

Size	$u^{*-}_{0.05}$ / $u^{*+}_{0.05}$	$u^{*-}_{0.01}$ / $u^{*+}_{0.01}$	$u^{*-}_{0.001}$ / $u^{*+}_{0.001}$
256 × 256	33.2824	33.7016	33.7677
	33.6447	33.2254	33.1593
384 × 384	33.5843	33.6222	33.6663
	33.3427	33.3048	33.2607
512 × 512	33.5541	33.5825	33.6156
	33.3729	33.3445	33.3114
1024 × 1024	33.5088	33.5230	33.5395
	33.4182	33.4040	33.3875
1536 × 1536	33.4937	33.5032	33.5142
	33.4333	33.4238	33.4128
3072 × 3072	33.4786	33.4833	33.4888
	33.4484	33.4437	33.4382

The critical values for NPCR and UACI are given in Tables 8 and 9 respectively. In Table 8 $N^*_{0.05}$, $N^*_{0.01}$, $N^*_{0.001}$ refer to the critical values for the rejection of the null hypothesis according to the significance levels of $\alpha = 0.05$, $\alpha = 0.01$, $\alpha = 0.001$. This means that if the value of NPCR for the two encrypted images is less than N^*_α then these two

TABLE 10. NPCR randomness test against critical values.

Size	NPCR value of the proposed scheme	0.05-level	0.01-level	0.001-level
256 × 256	99.6185	Pass	Pass	Pass
384 × 384	99.6195	Pass	Pass	Pass
512 × 512	99.6178	Pass	Pass	Pass
1024 × 1024	99.6168	Pass	Pass	Pass
1536 × 1536	99.6135	Pass	Pass	Pass
3072 × 3072	99.6103	Pass	Pass	Pass

TABLE 11. UACI randomness test against critical values.

Size	UACI value of the proposed scheme	$u^{*-}_{0.05}$ / $u^{*+}_{0.05}$	$u^{*-}_{0.01}$ / $u^{*+}_{0.01}$	$u^{*-}_{0.001}$ / $u^{*+}_{0.001}$
256 × 256	33.6068	Pass	Pass	Pass
384 × 384	33.3273	Fail	Pass	Pass
512 × 512	33.5280	Pass	Pass	Pass
1024 × 1024	33.4877	Pass	Pass	Pass
1536 × 1536	33.4585	Pass	Pass	Pass
3072 × 3072	33.4683	Pass	Pass	Pass

encrypted images are not sufficient random with an α-level of significance. One can see from Table 10 that the values of NPCR for all the sizes at all the levels of confidence for the proposed cipher fulfill the theoretical(critical) criterion of the NPCR randomness test. The critical value U^*_α for UACI is composed of two parts, i.e., U^{*-}_α, the left value and U^{*+}_α, the right value. Table 9 shows these values. If any value for the UACI of the proposed cipher is outside the interval $(U^{*-}_\alpha, U^{*+}_\alpha)$, the null hypothesis gets rejected. Table 11 shows that the values of UACI for all the sizes (except 384 × 384 at $\alpha = 0.05$) of the proposed cipher fulfill the critical values of the UACI randomness test.

F. PEAK SIGNAL-TO-NOISE RATIO (PSNR) ANALYSIS

The *PSNR* metric is used to measure the difference between the plain image and the encrypted image. A good image cipher is expected to create a maximum difference between the plain and encrypted images. The mathematical formulation of *PSNR* is given in (58):

$$\begin{cases} PSNR = 20log_{10}\left(255\big/\sqrt{MSE}\right) dB \\ MSE = \frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(P_0\left(i,j\right) - P_1(i,j)\right)^2 \end{cases} \tag{58}$$

where M and N are the width and height of the image respectively. $P_0(i, j)$ and $P_1(i, j)$ are the pixel values of the original and encrypted images respectively. Besides, Mean-Squared-Error (*MSE*) is the error/departure between the plain image and its encrypted version. The *MSE* is inversely proportional to *PSNR*. Larger the *MSE* value, the smaller will be the *PSNR* value, and the better the encryption security will be [50].

Table 12 highlights the *PSNR* values of our proposed scheme over our chosen images. The table shows two results, i.e., between the original plain image and decrypted image (*O-D*) and between the original plain image and the cipher image (*O-C*). As shown in the table, the *PSNR* values between (*O-D*) are always infinite (∞) meaning that the decrypted image is identical to the original image because of $MSE = 0$. This means that the proposed decrypted scheme produced 100% the same plain image without any loss. Further, in terms of the similarity between the original image and the encrypted image, the *PSNR* values of (*O-C*) for *3D-1* is 8.5477, *3D-2* is 8.6192, *3D-3* is 8.4988, *3D-4* is 8.5837 and *3D-6* is 8.5471. Table 12 also shows the comparison with other recent algorithms. It shows that the obtained values of our algorithm are the smallest in comparison to other algorithms [54]–[56]. So, our algorithm has a better encryption effect.

G. MEAN-ABSOLUTE-ERROR (MAE) ANALYSIS

The core objective of any encryption scheme is to maximize the difference between plain and encrypted images. The *MAE* is used for this purpose. The Mathematic formula of *MAE* is given in (59):

$$MAE = \frac{1}{M\times N}\sum\nolimits_{i=1}^{M}\sum\nolimits_{j=1}^{N}\left|C_{R,G,B}(i,j) - P_{R,G,B}(i,j)\right| \tag{59}$$

where P is for plain image and C is for cipher image, M is for the width and N is for the height of the image. The greater the obtained value of *MAE* is, the better the result will be. Table 13 shows the results of *MAE* produced by our proposed scheme and compares it with other schemes [6], [57], [58].

H. NOISE AND DATA LOSS ANALYSIS

In real-time scenarios, during the transmission of images from one point to another, some noise may include in the image. Sometimes, a portion of the image data may also lose. A good encryption scheme must deal with both noise and crop attacks. Figures 11(a) to 11(d) show the encrypted images (9 × 512 × 512) contaminated by Pepper & Salt noise with different noise densities, *i.e.*, 0.1, 0.2, 0.3 and 0.4 and Figures 11(e) to 11(h) show the corresponding decrypted images using our proposed scheme. The output shows that decrypted images can be easily recognized and most of the visual information is still intact.

Further Figures 12(a) to 12(d) plot the encrypted images (9 × 512 × 512), with data loss attacks of 0%, 25%, 50% and 75% respectively. After the decryption algorithm gets applied to these cropped cipher images, Figures 12(e) to 12(h) show the corresponding results.

It is obvious that the decrypted images from our proposed scheme still has most of the visual information. So, we are

TABLE 12. The-PSNR results between-the original images-and corresponding ciphered/decrypted-images: 'O –C' represents-the original-and ciphered images, and 'O-D' denotes the-original and-decrypted images.

Encryption Algorithm		3D-1	3D-2	3D-3	3D-4	3D-5	3D-6
Our Algorithm	PSNR(O-D)	Inf	Inf	Inf	Inf	Inf	Inf
	PSNR(O-C)	8.5477	8.6192	8.4988	8.4948	8.5837	8.5471
Ref. [54]	PSNR(O-D)	96.295					
	PSNR(O-C)	9.2322					
Ref. [55]	PSNR(O-C)	8.1717					
Ref. [56]	PSNR(O-C)	9.0486					

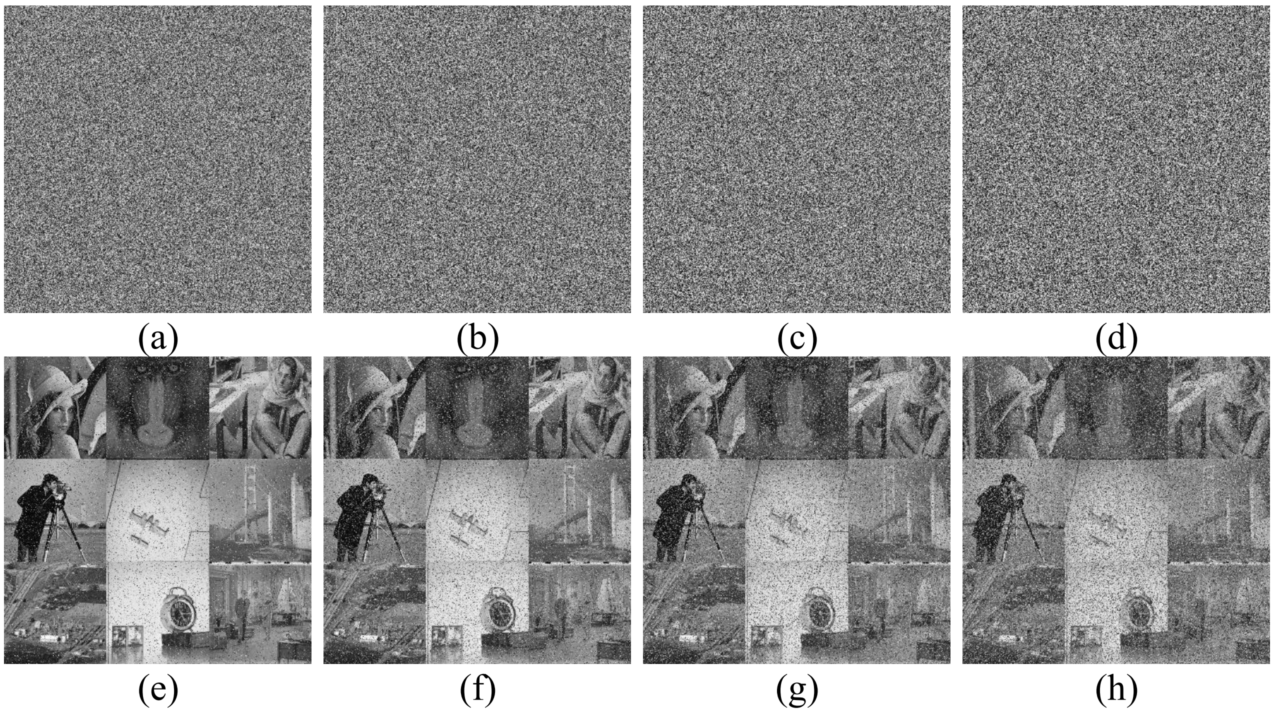


FIGURE 11. Pepper & Salt noise attacks: (a) Encrypted 3D-5 (9 × 512 × 512 size) image under adding Pepper & Salt noise with noise density 0.1; (b) Encrypted 3D-5 (9 × 512 × 512 size) image under adding Pepper & Salt noise with noise density 0.2; (c) Encrypted 3D-5 (9 × 512 × 512 size) image under adding Pepper & Salt noise with noise density 0.3; (d) Encrypted 3D-5 (9 × 512 × 512 size) image under adding Pepper & Salt noise with noise density 0.4; (e) Decrypted 3D-5 (9 × 512 × 512 size) image from (a); (f) Decrypted 3D-5 (9 × 512 × 512 size) image from (b); (g) Decrypted 3D-5 (9 × 512 × 512 size) image from (c); (h) Decrypted 3D-5 (9 × 512 × 512 size) image from (d).

TABLE 13. The MAE-results.

Encryption Algorithm	Test Images	No. of Images	MAE
Our Algorithm	*3D-1*	4x128x128	78.1019
	3D-2	9x128x128	77.5252
	3D-3	4x256x256	78.6274
	3D-4	4x512x512	78.5940
	3D-5	9x512x512	77.8662
	3D-6	9x1024x1024	77.8732
	Average		**78.0980**
Ref. [6]		512x512	76
Ref. [57]		256x256	70.9697
		512x512	75.0385
Ref. [58]		256x256	77.3500

justified in saying that the proposed scheme has an excellent capability to thwart any data loss attack during the transmission of images.

I. TIME COMPLEXITY ANALYSIS

This is a fact beyond any shadow of a doubt that the security of any image cipher is a primary concern of the cryptographers. In parallel to that, the performance of the cipher vis-à-vis time is not less important. The ciphers giving their result in relatively less time have more chances for their real-world application. Complying with these insights, this cipher has been built.

The proposed algorithms for the encryption and decryption have been developed and tested on Intel®Core^{TM} i7-3740QM Lenovo Thinkpad with CPU @ 2.70 GHz, 8GB Ram and 500GB Hard drive with Windows 10 Education operating system, and MATLAB R2018a. Table 14 shows the encryption/ decryption time of our proposed scheme and also conducts a comparison with other schemes [25], [26], [28], [29], [53].

Apart from the speed, the encryption throughput (ET) is another metric that deals with the amount of image encrypted in the unit time. The mathematical formula is

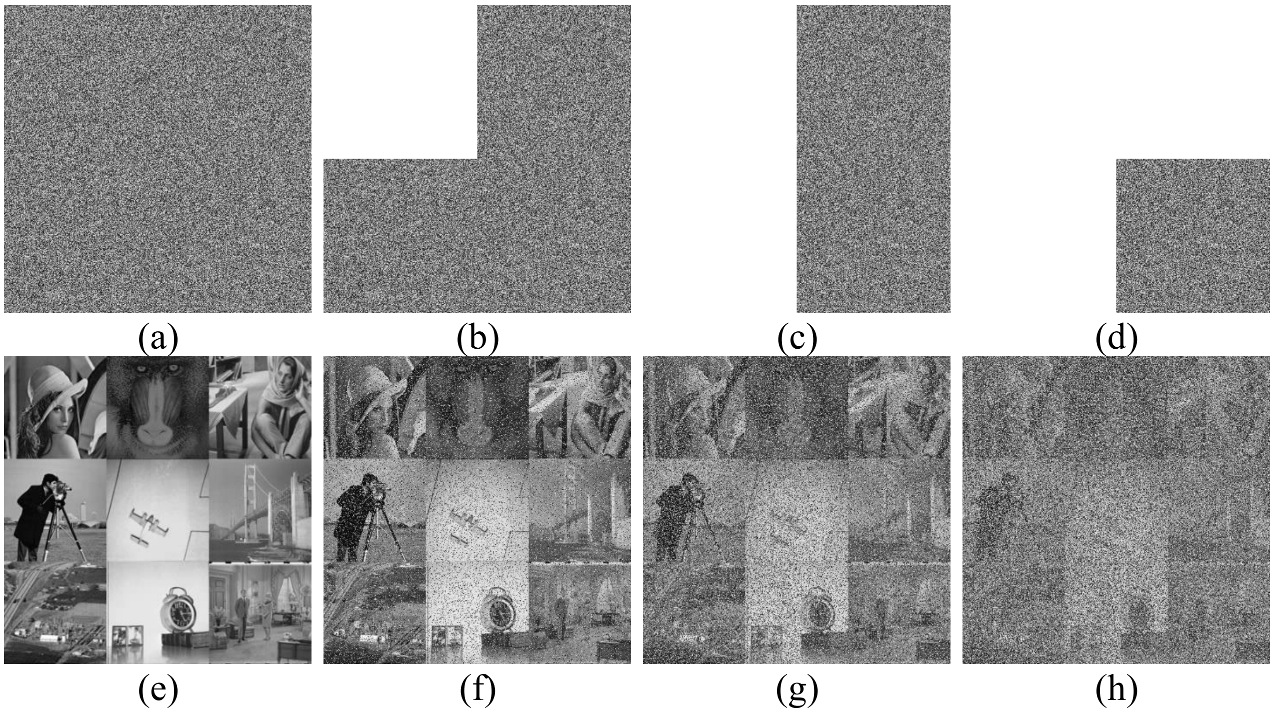


FIGURE 12. Data loss attack: (a) Encrypted Original 3D-5 (9 × 512 × 512 size) image; (b) Encrypted 3D-5 (9 × 512 × 512 size) image with 25% data loss; (c) Encrypted 3D-5 (9 × 512 × 512 size) image with 50% data loss; (d) Encrypted 3D-5 (9 × 512 × 512 size) image with 75% data loss; (e) Decrypted 3D-5 (9 × 512 × 512 size) image from (a); (f) Decrypted 3D-5 (9 × 512 × 512 size) image from (b); (g) Decrypted 3D-5 (9 × 512 × 512 size) image from (c); (h) Decrypted 3D-5 (9 × 512 × 512 size) image from (d).

TABLE 14. Encryption and decryption time execution in seconds.

Encryption Algorithm	Image	Image Size	No. of Images	Encryption Time (Sec)	Decryption Time (Sec)
Ours Algorithm	*3D-1*	256x256	4x128x128	0.30	0.28
	3D-2	384x384	9x128x128	0.41	0.33
	3D-3	512x512	4x256x256	1.10	0.93
	3D-4	1024x1024	4x512x512	9.97	8.57
	3D-5	1536x1536	9x512x512	21.68	18.75
	3D-6	3072x3072	9x1024x1024	66.85	55.79
Ref. [25]		256x256	4x128x128	0.38	
		512x512	4x256x256	0.90	
		1024x1024	4x512x512	3.65	
Ref. [53]		384x384	9x128x128	0.86	
		1536x1536	9x512x512	34.90	
		3072x3072	9x1024x1024	70.10	
Ref. [29]		384x384	9x128x128	0.55	
		1536x1536	9x512x512	22.40	
		3072x3072	9x1024x1024	45.20	
Ref. [26]		256x256		0.44	0.35
		512x512		0.85	0.76
		1024x1024		3.27	2.85
Ref. [28]		512x512		43.50	

mentioned in (60) as:

$$ET = \frac{Image_{Size}(Bit)}{Encryption_{Time}(Second)} \tag{60}$$

The ET of our proposed scheme is depicted in Table 15 along with a comparison with other existing schemes. The table shows that the proposed scheme has comparatively better ET than other existing schemes. In the literature,

TABLE 15. Encryption throughout of our proposed scheme and comparison with the existing schemes.

Encryption Algorithm	Image	Image Size	No. of Images	ET in Mbit/Sec
Ours Algorithm	*3D-1*	256x256	4x128x128	1.829
	3D-2	384x384	9x128x128	2.943
	3D-3	512x512	4x256x256	1.937
	3D-4	1024x1024	4x512x512	0.812
	3D-5	1536x1536	9x512x512	0.818
	3D-6	3072x3072	9x1024x1024	0.849
			Average	**1.531**
Ref. [50]		256x256		0.424
Ref. [59]				0.513
Ref. [60]				1.762

we couldn't find any MIE analyzing its ET. Therefore, we compare our MIE with the single image encryption schemes apropos ET.

VI. CONCLUSION

By using two chaotic maps and swapping operations of rows and columns in a 3D image space, a novel multiple images encryption schemes have been proposed. The chaotic data generated by the first map has been used for the project of scrambling whereas the data generated through the second map was used to create the diffusion effects in the proposed cipher. Before the scrambling process is launched, the input images are stacked to form a 3D image. In each iteration, two images are selected randomly from the pile of images. From these two selected images, two rows are selected randomly from each image and they are swapped with each other. In the same fashion, two columns selected randomly are swapped with each other. The images, rows, and columns have been selected in a purely arbitrary manner thus boosting the randomization process which in turn heightens the security effects for the cipher. It is to be noted that the three streams given by the first map have been used in the swapping/scrambling process. In each paired selection of the images, rows, and columns, the chaotic data of the streams have been used from both ends. The scrambled 3D image is further XORed with the random data given by the second chaotic map. To incorporate the plaintext sensitivity in the proposed cipher, SHA-256 hash codes have been employed to temper the initial values and the system parameters of the maps. Besides, a 256-bit user key has also been used in the cipher to increase the key space. Six different sizes of the images have been used to demonstrate the capability of the cipher. The simulation and the sweeping security analysis expressly indicate the security, defiance to the varied threats, and the real-world applicability for the proposed image cipher.

ACKNOWLEDGMENT

(Muhammad Hanif and Rizwan Ali Naqvi are co-first authors.)

REFERENCES

[1] L. Xu, Z. Li, J. Li, and W. Hua, "A novel bit-level image encryption algorithm based on chaotic maps," *Opt. Lasers Eng.*, vol. 78, pp. 17–25, Mar. 2016.

[2] Z. Parvin, H. Seyedarabi, and M. Shamsi, "A new secure and sensitive image encryption scheme based on new substitution with chaotic function," *Multimedia Tools Appl.*, vol. 75, no. 17, pp. 10631–10648, Sep. 2016.

[3] Z. Bashir, T. Rashid, and S. Zafar, "Hyperchaotic dynamical system based image encryption scheme with time-varying delays," *Pacific Sci. Rev. A, Natural Sci. Eng.*, vol. 18, no. 3, pp. 254–260, Nov. 2016.

[4] A. Y. Niyat and M. H. Moattar, "Color image encryption based on hybrid chaotic system and DNA sequences," *Multimedia Tools Appl.*, vol. 79, pp. 1497–1518, Nov. 2019.

[5] X. Wang, Y. Wang, X. Zhu, and C. Luo, "A novel chaotic algorithm for image encryption utilizing one-time pad based on pixel level and DNA level," *Opt. Lasers Eng.*, vol. 125, Feb. 2020, Art. no. 105851.

[6] F. Masood, J. Ahmad, S. A. Shah, S. S. Jamal, and I. Hussain, "A novel hybrid secure image encryption based on Julia set of fractals and 3D lorenz chaotic map," *Entropy*, vol. 22, no. 3, pp. 274–303, 2020.

[7] Y. Luo, J. Yu, W. Lai, and L. Liu, "A novel chaotic image encryption algorithm based on improved baker map and logistic map," *Multimedia Tools Appl.*, vol. 78, pp. 22023–22043, Apr. 2019.

[8] M. A. B. Farah and A. F. T. Farah, "An image encryption scheme based on a new hybrid chaotic map and optimized substitution box," *Nonlinear Dyn.*, vol. 99, pp. 3041–3064, Dec. 2019.

[9] H. Zhu, Y. Zhao, and Y. Song, "2D logistic-modulated-sine-coupling-logistic chaotic map for image encryption," *IEEE Access*, vol. 7, pp. 14081–14098, 2019.

[10] Q. Lu, C. Zhu, and X. Deng, "An efficient image encryption scheme based on the LSS chaotic map and single S-box," *IEEE Access*, vol. 8, pp. 25664–25679, 2020.

[11] Y. Hu, C. Zhu, and Z. Wang, "An improved piecewise linear chaotic map based image encryption algorithm," *Sci. World J.*, vol. 2014, pp. 1–7, Jan. 2014.

[12] I. Shatheesh Sam, P. Devaraj, and R. S. Bhuvaneswaran, "An intertwining chaotic maps based image encryption scheme," *Nonlinear Dyn.*, vol. 69, no. 4, pp. 1995–2007, Sep. 2012.

[13] M. Babaei, "A novel text and image encryption method based on chaos theory and DNA computing," *Natural Comput.*, vol. 12, no. 1, pp. 101–107, Mar. 2013.

[14] A. Jain and N. Rajpal, "A robust image encryption algorithm resistant to attacks using DNA and chaotic logistic maps," *Multimedia Tools Appl.*, vol. 75, no. 10, pp. 5455–5472, May 2016.

[15] Q. Zhang, L. Liu, and X. Wei, "Improved algorithm for image encryption based on DNA encoding and multi-chaotic maps," *AEU-Int. J. Electron. Commun.*, vol. 68, no. 3, pp. 186–192, Mar. 2014.

[16] R. Boriga, A. C. Dăscălescu, and I. Priescu, "A new hyperchaotic map and its application in an image encryption scheme," *Signal Process., Image Commun.*, vol. 29, no. 8, pp. 887–901, Sep. 2014.

[17] O. Mirzaei, M. Yaghoobi, and H. Irani, "A new image encryption method: Parallel sub-image encryption with hyper chaos," *Nonlinear Dyn.*, vol. 67, no. 1, pp. 557–566, Jan. 2012.

[18] H. Liu and X. Wang, "Color image encryption using spatial bit-level permutation and high-dimension chaotic system," *Opt. Commun.*, vol. 284, nos. 16–17, pp. 3895–3903, Aug. 2011.

[19] C. Li and K.-T. Lo, "Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal Process.*, vol. 91, no. 4, pp. 949–954, Apr. 2011.

[20] S. Li, C. Li, G. Chen, D. Zhang, and N. G. Bourbakis, "A general cryptanalysis of permutation-only multimedia encryption algorithms," in *Proc. 24rd Annu. Int. Cryptol. Conf. (CRYPTO)*, Santa Barbara, CA, USA, 2004.

[21] J. Wu, X. Liao, and B. Yang, "Cryptanalysis and enhancements of image encryption based on three-dimensional bit matrix permutation," *Signal Process.*, vol. 142, pp. 292–300, Jan. 2018.

[22] F. Özkaynak and A. B. Özer, "Cryptanalysis of a new image encryption algorithm based on chaos," *Optik*, vol. 127, no. 13, pp. 5190–5192, Jul. 2016.

[23] M. Khan and T. Shah, "An efficient chaotic image encryption scheme," *Neural Comput. Appl.*, vol. 26, no. 5, pp. 1137–1148, Jul. 2015.

[24] X. Chai, X. Fu, Z. Gan, Y. Lu, and Y. Chen, "A color image cryptosystem based on dynamic DNA encryption and chaos," *Signal Process.*, vol. 155, pp. 44–62, Feb. 2019.

[25] Z. Shao, X. Liu, Q. Yao, N. Qi, Y. Shang, and J. Zhang, "Multiple-image encryption based on chaotic phase mask and equal modulus decomposition in quaternion gyrator domain," *Signal Process., Image Commun.*, vol. 80, Feb. 2020, Art. no. 115662.
[26] K. A. K. Patro and B. Acharya, "A novel multi-dimensional multiple image encryption technique," *Multimedia Tools Appl.*, vol. 79, pp. 15–25, Jan. 2020.
[27] A. A. Karawia, "Encryption algorithm of multiple-image using mixed image elements and two dimensional chaotic economic map," *Entropy*, vol. 20, no. 10, pp. 80–101, 2018.
[28] X. Zhang and X. Wang, "Multiple-image encryption algorithm based on DNA encoding and chaotic system," *Multimedia Tools Appl.*, vol. 78, no. 6, pp. 7841–7869, Mar. 2019.
[29] R. Enayatifar, F. G. Guimarães, and P. Siarry, "Index-based permutation-diffusion in multiple-image encryption using DNA sequence," *Opt. Lasers Eng.*, vol. 115, pp. 131–140, Apr. 2019.
[30] Z. Tang, J. Song, X. Zhang, and R. Sun, "Multiple-image encryption with bit-plane decomposition and chaotic maps," *Opt. Lasers Eng.*, vol. 80, pp. 1–11, May 2016.
[31] X. Zhang and X. Wang, "Multiple-image encryption algorithm based on mixed image element and permutation," *Opt. Lasers Eng.*, vol. 92, pp. 6–16, May 2017.
[32] S. Liansheng, Z. Xiao, H. Chongtian, T. Ailing, and A. K. Asundi, "Silhouette-free interference-based multiple-image encryption using cascaded fractional Fourier transforms," *Opt. Lasers Eng.*, vol. 113, pp. 29–37, Feb. 2019.
[33] M. Y. M. Parvees, J. A. Samath, and B. P. Bose, "Secured medical images—A chaotic pixel scrambling approach," *J. Med. Syst.*, vol. 40, no. 11, pp. 1–11, Nov. 2016.
[34] K.-W. Wong, B. S.-H. Kwok, and C.-H. Yuen, "An efficient diffusion approach for chaos-based image encryption," *Chaos, Solitons Fractals*, vol. 41, no. 5, pp. 2652–2663, Sep. 2009.
[35] J.-X. Chen, Z.-L. Zhu, C. Fu, and H. Yu, "An improved permutation-diffusion type image cipher with a chaotic orbit perturbing mechanism," *Opt. Express*, vol. 21, no. 23, p. 27873, 2013.
[36] R. Ye, Y. Xi, and Y. Ma, "A chaotic image encryption scheme using swapping based confusion approach," in *Proc. 1st IEEE Int. Conf. Comput. Commun. Internet (ICCCI)*, Oct. 2016, pp. 374–377.
[37] X. Chai, Z. Gan, and M. Zhang, "A fast chaos-based image encryption scheme with a novel plain image-related swapping block permutation and block diffusion," *Multimedia Tools Appl.*, vol. 76, no. 14, pp. 15561–15585, Jul. 2017.
[38] J.-X. Chen, Z.-L. Zhu, C. Fu, and H. Yu, "A fast image encryption scheme with a novel pixel swapping-based confusion approach," *Nonlinear Dyn.*, vol. 77, no. 4, pp. 1191–1207, Sep. 2014.
[39] C. Fu, G.-Y. Zhao, M. Gao, and H.-F. Ma, "A chaotic symmetric image cipher using a pixel-swapping based permutation," in *Proc. IEEE Int. Conf. IEEE Region 10 (TENCON)*, Oct. 2013, pp. 1–6.
[40] H. R. Amani and M. Yaghoobi, "A new approach in adaptive encryption algorithm for color images based on DNA sequence operation and hyper-chaotic system," *Multimedia Tools Appl.*, vol. 78, no. 15, pp. 21537–21556, Aug. 2019.
[41] C. Yu, J. Li, X. Li, X. Ren, and B. B. Gupta, "Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram," *Multimedia Tools Appl.*, vol. 77, no. 4, pp. 4585–4608, 2018.
[42] A. Patel and M. Parikh, "Multiple image encryption using chaotic map and DNA computing," *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 4, no. 4, pp. 1395–1400, 2018.
[43] X. Zhang and X. Wang, "Multiple-image encryption algorithm based on mixed image element and chaos," *Comput. Elect. Eng.*, vol. 1, no. 1, pp. 1–13, 2017.
[44] W. Liu, K. Sun, and C. Zhu, "A fast image encryption algorithm based on chaotic map," *Opt. Lasers Eng.*, vol. 84, pp. 26–36, Sep. 2016.
[45] X. Wu, K. Wang, X. Wang, H. Kan, and J. Kurths, "Color image DNA encryption using NCA map-based CML and one-time keys," *Signal Process.*, vol. 148, pp. 272–287, Jul. 2018.
[46] X. Liao, A. Kulsoom, and S. Ullah, "A modified (Dual) fusion technique for image encryption using SHA-256 hash and multiple chaotic maps," *Multimedia Tools Appl.*, vol. 75, no. 18, pp. 11241–11266, 2016.
[47] J. S. Khan and J. Ahmad, "Chaos based efficient selective image encryption," *Multidimensional Syst. Signal Process.*, vol. 30, no. 2, pp. 943–961, 2019.
[48] A. Kulsoom, D. Xiao, and S. A. Abbas, "An efficient and noise resistive selective image encryption scheme for gray images based on chaotic maps and DNA complementary rules," *Multimedia Tools Appl.*, vol. 75, no. 1, pp. 1–23, 2016.
[49] X. Liao, M. A. Hahsmi, and R. Haider, "An efficient mixed inter-intra pixels substitution at 2bits-level for image encryption technique using DNA and chaos," *Optik-Int. J. Light Electron Opt.*, vol. 153, pp. 117–134, 2018.
[50] N. Iqbal, S. Abbas, M. A. Khan, T. Alyas, A. Fatima, and A. Ahmad, "An RGB image cipher using chaotic systems, 15-puzzle problem and DNA computing," *IEEE Access*, vol. 7, pp. 174051–174071, 2019.
[51] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons Fractals*, vol. 21, no. 3, pp. 749–761, Jul. 2004.
[52] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.
[53] R. Enayatifar, A. H. Abdullah, I. F. Isnin, A. Altameem, and M. Lee, "Image encryption using a synchronous permutation-diffusion technique," *Opt. Lasers Eng.*, vol. 90, pp. 146–154, Mar. 2017.
[54] C. Zhu, "A novel image encryption scheme based on improved hyperchaotic sequences," *Opt. Commun.*, vol. 285, no. 1, pp. 29–37, Jan. 2012.
[55] B. Norouzi, S. M. Seyedzadeh, S. Mirzakuchaki, and M. R. Mosavi, "A novel image encryption based on hash function with only two-round diffusion process," *Multimedia Syst.*, vol. 20, no. 1, pp. 45–64, Feb. 2014.
[56] N. Taneja, B. Raman, and I. Gupta, "Combinational domain encryption for still visual data," *Multimedia Tools Appl.*, vol. 59, no. 3, pp. 775–793, Aug. 2012.
[57] H. Liu, B. Zhao, and L. Huang, "Quantum image encryption scheme using Arnold transform and S-box scrambling," *Entropy*, vol. 21, no. 4, pp. 1–14, 2019.
[58] M. Khan, "A novel image encryption scheme based on multiple chaotic S-boxes," *Nonlinear Dyn.*, vol. 82, nos. 1–2, pp. 527–533, Oct. 2015.
[59] T. Hu, Y. Liu, L.-H. Gong, and C.-J. Ouyang, "An image encryption scheme combining chaos with cycle operation for DNA sequences," *Nonlinear Dyn.*, vol. 87, no. 1, pp. 51–66, Jan. 2017.
[60] Y. Liu, J. Wang, J. Fan, and L. Gong, "Image encryption algorithm based on chaotic system and dynamic S-boxes composed of DNA sequences," *Multimedia Tools Appl.*, vol. 75, no. 8, pp. 4363–4382, Apr. 2016.

MUHAMMAD HANIF received the B.S. degree in information technology from the University of Malakand, Pakistan, and the M.S. degree in information technology from SEECS, NUST, Islamabad, Pakistan. He is currently pursuing the Ph.D. degree in computer science with NCBA&E Lahore, Pakistan. He is currently working as an Assistant Professor with the Department of Computer Science, Bahria University, Lahore Campus, Pakistan. He was with various academic institutions and has supervised numerous bachelor's and master's students from the past eight years. His current research interests include images cryptography, computer graphics, networking, cloud computing, and the Internet of Things.

RIZWAN ALI NAQVI (Member, IEEE) received the B.S. degree in computer engineering from COMSATS University, Pakistan, in 2008, the M.S. degree in electrical engineering from Karlstad University, Sweden, in 2011, and the Ph.D. degree in electronics and electrical engineering from Dongguk University, South Korea, in 2018. From 2011 to 2012, he was a Lecturer with the Computer Science Department, Sharif College of Engineering and Technology, Pakistan. He joined the Faculty of Engineering and Technology, The Superior College, Pakistan, as a Senior Lecturer, in 2012. After his Ph.D. degree, he has worked as a Postdoctoral Researcher with Gachon University, South Korea, from 2018 to 2019. He is currently working as an Assistant Professor with Sejong University, South Korea. His research interests include gaze tracking, biometrics, computer vision, artificial intelligence, machine learning, deep learning, and medical imaging analysis.

SAGHEER ABBAS received the M.Phil. degree in computer science and the Ph.D. degree from the School of Computer Science, NCBA&E, Lahore, Pakistan. He has been teaching graduate and undergraduate students in computer science and engineering for the past eight years. He is currently an Assistant Professor with the School of Computer Science, NCBA&E. He has published about 60 research articles in international journals and reputed international conferences. His current research interests include cloud computing, the IoT, intelligent agents, image processing, and cognitive machines with various publications in international journals and conferences.

MUHAMMAD ADNAN KHAN received the B.S. and M.Phil. degrees from International Islamic University, Islamabad, Pakistan, and the Ph.D. degree from ISRA University, Pakistan. He is currently working as an Assistant Professor with the Department of Computer Science, Lahore Garrison University, Lahore, Pakistan. Before joining Lahore Garrison University, he has worked in various academic and industrial roles in Pakistan. He has been teaching graduate and undergraduate students in computer science and engineering for the past 12 years. He is also guiding four Ph.D. scholars and four M.Phil. scholars. He has published about 150 research articles in International Journals as well as reputed International Conferences. His research interests include MUD, image processing and medical diagnosis, channel estimation in multi-carrier communication systems using soft computing with various publications in journals and conferences of international repute. He received the Scholarship Award from the Punjab Information and Technology Board, Government of Punjab, Pakistan, for his B.S. and M.Phil. degrees; and the Scholarship Award from the Higher Education Commission, Islamabad, for his Ph.D. degree.

NADEEM IQBAL received the M.Phil. degree in computational science and engineering from NUST, Islamabad, Pakistan. He is currently working as an Assistant Professor with the School of Computing and Information Sciences, Imperial College of Business Studies (ICBS), Lahore, Pakistan. Prior to joining the ICBS, he has worked in various academic institutions and has guided numerous undergrad and masters students. His research interests include images cryptography, computer graphics, and philosophy of mathematics.